TP 10414E

Final Report

KNOWLEDGE-BASED DISPATCHER'S ASSISTANT

Technical Report: AAI-90-3

September 1990

Prepared for:

Transportation Development Centre
Policy and Coordination Group
Transport Canada

Prepared by:

Applied AI Systems, Inc.
Suite 500, Gateway Business Park
340 March Road
KANATA, Ontario, Canada K2K 2E4
Tel. (613) 592-3030
Fax. (613) 592-2333

(C) Applied AI Systems, Inc.

92-00968/

CA1
DTC800-
90
K52
458971

41.D

TP 10414E

Final Report

KNOWLEDGE-BASED DISPATCHER'S ASSISTANT

Technical Report: AAI-90-3

September 1990

Prepared for:

Transportation Development Centre
Policy and Coordination Group
Transport Canada

Prepared by:

Applied AI Systems, Inc.
Suite 500, Gateway Business Park
340 March Road
KANATA, Ontario, Canada K2K 2E4
Tel. (613) 592-3030
Fax. (613) 592-2333

(C) Applied AI Systems, Inc.

28
[illegible]

458971

DDN 453816

TP 10414E

KNOWLEDGE-BASED DISPATCHER'S ASSISTANT

Written by:

Takashi Gomi
Ann Griffith
Karl Mayer
Use Oppacher

Applied AI Systems, Inc.

For:

Transportation Development Center
Policy and Coordination Group
Transport Canada

September 1990

The contents of this report reflect the views of the authors and Applied AI Systems, Inc. and not necessarily the official view or opinions of the Transportation Development Centre of Transport Canada.

Un sommaire en français de ce rapport est inclus avant la table des matieres.

Project Team

The following are the members of the development team. The roles that they play are listed beside their names.

Takashi Gomi	Project Manager Knowledge Engineer
Ann Griffith	Project Administrator Junior AI Analyst/Programmer
Use Oppacher	Junior AI Programmer
Karl Mayer	Junior AI Programmer

Transport Canada Transports Canada

PUBLICATION DATA FORM

1. Transport Canada Publication No.
TP 1041E

2. Project No.
6900-6901

3. Recipient's Catalogue No.

4. Title and Subtitle
KNOWLEDGE-BASED DISPATCHER'S ASSISTANT

5. Publication Date
August 1990

6. Performing Organization Document No.

7. Author(s)
Takashi Gomi, Ann Griffith, Karl Mayer,
Use Oppacher

8. Transport Canada File No.
D1465-509

9. Performing Organization Name and Address
Applied AI Systems, Inc.
340 March Road, Suite 500
Kanata, Ontario
K2K 2E4

10. DSS File No.
14SD.T8200-7-7502

11. DSS or Transport Canada Contract No.
14SD.T8200-7-7502

12. Sponsoring Agency Name and Address
Transportation Development Centre
200 René Lévesque Blvd., West
West Tower, Suite 601
Montreal, Québec
H2Z 1X4

13. Type of Publication and Period Covered
Technical Report
AAI-90-3

14. Project Officer
Lewis Sabounghi

15. Supplementary Notes (Funding programs, titles of related publications, etc.)

16. Abstract

This report describes a prototype knowledge-based system designed to assist dispatchers in assigning staff and equipment for the delivery of various liquid gas products to a large customer base. The dispatcher must ensure that customer requirements are met on time, and that drivers, trucks, and trailers are used efficiently. A series of interviews with the human performing the task and data extracted from actual operations were used to construct the system's knowledge base and data structures.

The system allows the dispatcher to select, in a highly interactive manner, from various scheduling options. The primary output of the system is a delivery order for each assigned driver. The system was demonstrated to staff at the dispatch site and received a positive response with support for continued development. It is recommended that further work be done to integrate the knowledge-based system with the existing computerized system, to generalize the system developed and make the system architecture more generic, and to conduct further engineering sessions. Voice recognition and voice synthesis subsystems were used as parallel interfaces to the conventional keyboard/screen pair, as well as an automated dictation system for filling in reports and notes without using a keyboard.

17. Key Words
Expert systems, knowledge-based systems, user interface, voice technology, AI in transportation

18. Distribution Statement
Limited number of copies available from the Transportation Development Centre on request.

19. Security Classification (of this publication)
Unclassified

20. Security Classification (of this page)
Unclassified

21. Declassification (date)

22. No. of Pages
74

23. Price

TDC/CDT 79-005
Rev 88

Canada

Transports Canada Transport Canada

FORMULE DE DONNÉES POUR PUBLICATION

1. N° de la publication de Transports Canada
TP 10414E

2. N° de l'étude
6900-6901

3. N° de catalogue du destinataire

4. Titre et sous-titre
KNOWLEDGE-BASED DISPATCHER'S ASSISTANT

5. Date de la publication
août 1990

6. N° du document de l'organisme exécutant

7. Auteur(s)
Takashi Gomi, Ann Griffith, Karl Mayer,
Use Oppacher

8. N° de dossier — Transports Canada
D1465-509

9. Nom et adresse de l'organisme exécutant
Applied AI Systems, Inc.
340 March Road, Suite 500
Kanata (Ontario)
K2K 2E4

10 N° de dossier — ASC
14SD.T8200-7-7502

11. N° de contrat — ASC ou Transports Canada
14SD.T8200-7-7502

12 Nom et adresse de l'organisme parrain
Centre de développement des transports
200 ouest, boul. René-Lévesque
Tour ouest, suite 601
Montréal (Québec) H2Z 1X4

13. Genre de publication et période visée
Rapport technique
AAI-90-3

14. Agent de projet
Lewis Sabounghi

15. Remarques additionnelles (Programmes de financement, titres de publications connexes, etc.)

16. Résumé

Le présent rapport décrit la recherche faite sur le prototype d'un système expert à base de connaissances destiné à aider les régulateurs à prendre des décisions concernant l'affectation de personnel et de matériel pour effectuer des livraisons de gaz liquides à une très vaste clientèle. Les critères à respecter sont que les livraisons doivent se faire en temps voulu et que l'efficacité des acheminements soit aussi élevée que possible. Pour construire la base de connaissances et structurer les données, les chercheurs se sont inspirés des explications données par un régulateur lors d'interviews effectuées dans son milieu de travail, et de données connaissances tirées puisées des registres d'exploitation dans ce même milieu.

Le système mis au point permet au régulateur de choisir interactivement une grille des livraisons parmi plusieurs proposées par le système et, à partir de celle-ci, de procéder aux affectations nécessaires. La démonstration du système faite en milieu de travail et devant le personnel directement intéressé a reçu un bon accueil de ce dernier, qui s'est ensuite prononcé en faveur de la continuation de la recherche. Les recommandations formulées à cet effet sont d'intégrer le système dans le programme de livraisons automatiques existant, de le généraliser, d'en rendre l'architecture plus générique et de mener plusieurs autres séances de développement. Outre le dispositif d'entrée à clavier alphanumérique classique, des dispositifs de reconnaissance et de synthèse vocales sont proposées, ainsi qu'un système de dictée automatique capable de produire des rapports et des messages sans le secours d'un clavier.

17. Mots clés
Systèmes experts, systèmes à base de connaissances, interface utilisateur, technologie des commandes vocales, IA en transport.

18. Diffusion
Le Centre de développement des transports dispose d'un nombre limité d'exemplaires.

19. Classification de sécurité (de cette publication)	20. Classification de sécurité (de cette page)	21. Déclassification (date)	22. Nombre de pages	23. Prix
Non classifiée	Non classifiée	-	74	-

CDT/TDC 79-005
Rev 88

Canada

Acknowledgements

The authors wish to acknowledge the participation of Canadian Liquid Air Limited in the development of this contract. Mr. Dennis Ogilvie was the senior dispatcher who was interviewed for his skills in dispatching the drivers for the distribution of liquid gases.

Summary

1. Introduction

Transportation and related activities play an important role in the lives of all Canadians. With the expansive geography of our country, the distance required to travel to a destination is often a major consideration. As the infrastructure of Canada's transportation system grows, the problems of managing the system grow also. There is always the requirement to move the largest volume of goods and people at the lowest cost while still maintaining a safe transportation environment.

Trucking and moving goods by large vehicles on Canada's roads is a necessity. In late 1985 and early 1986 discussions took place between the Transportation Development Centre (TDC) of Transport Canada and Applied AI Systems (AAI) about the types of knowledge and the expertise used in the decision-making process of dispatching trucks and their drivers. Some initial investigations were conducted by visiting a dispatcher responsible for the dispatching for delivery of liquid gases, Mr. Dennis Ogilvie at Canadian Liquid Air Limited (CLA) in Hamilton, Ontario. This preliminary visit indicated that the area of dispatching was an appropriate one for the application of knowledge-based technology.

The dispatcher responsible for dispatching the drivers of bulk tankers is required, as is any dispatcher, to consider many factors, to make quick judgments and then to integrate the results. An advisory knowledge-based system that assisted with some aspects of these tasks would give the dispatcher more time to concentrate on the more complex, abstract and challenging tasks beyond the capability of existing computer system technology.

Dispatchers are constantly under pressure to complete their daily assignments of tractor, trailer and list of customers for each driver in an environment where the requirements and resources are subject to changes that are often beyond their control. In making the assignments, the dispatcher must consider the distance between customers and the distance from the plant. The trailer must be returned in sufficient time for it to be refilled with product and ready for the next shift of drivers.

The dispatcher must also cope with the unexpected emergencies that seem to arise each day. A breakdown of equipment, a driver phoning in sick, or a customer needing a product sooner than expected can happen on any given day, and these situations must be dealt with while still providing the already scheduled service to the customer.

In June 1987, Applied AI Systems, Inc (AAI) submitted an unsolicited proposal to the Department of Supply and Services Canada entitled "Development and Demonstration of Knowledge-Based Vehicle Dispatcher's Assistant". In August 1987 a contract was awarded to AAI to begin development of the prototype system.

The project goal was to contribute to the improved safety and productivity of highway freight vehicles by providing the means for a consistent and improved level of decision making in the dispatching process and to introduce expert system technology to surface technology systems in Canada. It was hoped the developed prototype might assist less experienced dispatchers in performing the complex dispatching task.

2. Domain

The dispatcher of the Bulk Distribution Department at Canadian Liquid Air is responsible for assuring that the companies drivers, vehicles and shipment destinations are coordinated in an optimal fashion. The dispatcher is also responsible for ensuring that customer needs are met in a timely manner. For example, hospitals cannot provide adequate health services without a guaranteed supply of oxygen. To run out of product could be very costly for a customer in terms of production time lost and time wasted by personnel and equipment, and in some cases could endanger human lives. The need to minimize delivery costs and maximize volume must be balanced with concerns for safety and the varying needs of the customers. All this places considerable stress on the dispatcher, who is already contending with many variables and uncertainties, many of which are beyond his control.

Because of the number of factors to consider, and because situations may change from day to day and hour to hour, dispatching can be a complex problem requiring considerable time and skill. The greater the number of drivers, tractors, trailers, types of gases, and customers the dispatcher must consider, the more time-consuming the task becomes and the greater the number of unexpected situations that may arise.

In addition to the variables, which may change each day, there are also certain requirements that remain stable and must be met with each daily assignment. The bulk carriers of liquid gases must conform to the regulations of the Transportation of Dangerous Goods Act. This includes ensuring that each vehicle is equipped with the proper placarding identifying the product being transported and ensuring that each driver completes the required paperwork. Weight restrictions placed on large vehicles by the Transportation department of the province must be followed. Union agreements that have been negotiated between management and the drivers must be honoured. For example, drivers must not be driving for more than six hours in any twenty-four-hour period. All of these regulations impose restrictions that must be taken into consideration when the dispatch assignments are done.

In addition, there are other factors that are not "rules" or "laws" but nonetheless affect the assignment process. In particular, the customer must be taken into consideration. There are sometimes limitations imposed by the customer. For example, perhaps the customer's yard is small and can only accommodate certain trailers. Perhaps the firm's gate must be locked for security reasons and delivery can only be made at certain times of the day. If the liquid gas in the tank needs to be under high pressure to meet the customer's needs, then only certain trailers can pump above that pressure, thus limiting the trailers that can deliver to that site. These restrictions and considerations are additional factors that must be taken into consideration when completing the dispatching process.

Canadian Liquid Air handles mostly liquid gases and distribute the product to customers in central and western Ontario, northern Ontario and as far east as Kingston. Their geographical coverage is broken into zones. Oxygen, Carbon Dioxide, and Nitrogen have seven zones, and Argon has four. For any given assignment, drivers are assigned to customers within a zone and the goal is to have the trailer leave with a full tank and return to Hamilton empty.

A meter on the trailer records the exact amount of product delivered to each customer. Each driver is responsible for giving this record to the Bulk Distribution Department when he returns from his deliveries and approximately three hours is spent each morning inputting this record. This data is then used by the computerized prediction system currently in place. Based on the customers' past consumption, the prediction system provides a daily printout of customers that will require delivery for the following eleven days. A customer's name starts to appear on the printout four days before it is critical that delivery take place. This gives the dispatcher some flexibility in planning his schedule so that he can group customers in the same area together for most efficient delivery on any given day. From the printout, the dispatcher makes a list of customers for each gas to be delivered, and completes his assignment.

The customers have the option of how their delivery will be determined. The majority of customers are on the computerized prediction system but some elect other means. Some customers choose to be responsible for observing the level of their tank and calling when they require delivery (call-in customer). This is preferred by customers whose irregular consumption makes it impossible for an outsider to know when sudden large increases may occur for a short period of time. Other customers receive delivery on a fixed schedule, for example, every Monday and Thursday or every three days. The type of delivery arrangement is worked out with the customers to best meet their needs and to ensure that they will always have a constant supply of the product.

3. Tool Selected

The selected tool for the implementation of the project was NEXPERT Object, an expert system development shell developed by Neuron Data, in Palo Alto, California. Although it was still a relatively new tool on the market at the beginning of the project, the flexibility and power of the tool was already evident. The development was conducted on a 386 IBM compatible PC.

NEXPERT is a hybrid rule-based and object-based tool that provides a powerful reasoning mechanism for linking facts or observations to assertions or actions. The developers have recognized the need to integrate with existing computer environments and thus have developed links for NEXPERT to many other languages, and the capability to retrieve information and update existing databases. NEXPERT can be embedded in existing applications or it can be the controlling program and execute other external routines.

The development environment in NEXPERT provides a graphical, spreadsheet-like format for the writing of rules, in the Rule Editor. The Rule Network automatically generates a graphic representation of the knowledge base and its control structure. This visual representation of the links between the concepts and the rules assists in the debugging of knowledge bases and provides a clearer image of the structure of the knowledge base. The reasoning kernel within NEXPERT allows for forward and backward chaining, automatic goal generation, multiple inheritance, pattern matching, and direct calls to and from external routines.

During the course of the project there were several upgrades, enhancements and announcements of new products. The developers experienced some difficulties related to the capabilities of the tool. For example, the Protected Mode version of the runtime did not become available until after considerable time had been spent trying to resolve problems which were discovered to be related to memory and solved by the release of the Protected Mode.

The Runtime delivery system is a different interface from the Development version. NORT (NEXPERT Object Runtime) interprets questions, forms and reports written in the Screen Definition Language for user input and output. A batch control file (.RTD) volunteers data, suggests hypotheses, displays the data entry forms, and shows report forms or files when it is finished.

One of the reasons for the selection of NEXPERT Object as the tool for the project was the emphasis that the developers of NEXPERT have placed on the user interface. Since the development of the interface is one of the most important factors in determining the success or failure of an expert system, it was important to select a tool that allowed, and provided the facilities for, a flexible interface. This will be discussed in more detail in the following section on the user interface. NEXPERT provided the capabilities to build a custom interface with many of the features that have been identified as important for acceptance by the user.

4. Project Description

The completed prototype consists of a knowledge base made up of two files, data that is read from external files and updated by the program, and some external routines called from within NEXPERT.

The two files that make up the knowledge base are loaded consecutively during execution. They contain a collection of rules and the data structures required to run a session. The control mechanism, called the Knowledge Processor, intelligently and dynamically browses through the knowledge base. During the design and construction of the knowledge base, links are established between the rules. These links reflect the actual structure of the knowledge and determine the execution of the program. Rules are written in an "If...Then" format and an example rule is as follows:

```
If  there is no evidence of B10_no_trail_change
    And Execute "voice.exe"
    (@TYPE=EXE;@WAIT=TRUE;@STRING="6";)
    And there is evidence of trail_change
    And there is evidence of <|trailers|>.avail
Then B20-change_trail
    is confirmed.
    And Create Object <|trailers|> |tr_avail|
    And TRUE is assigned to avail_trail
```

The data require by the knowledge base is contained in database files with an extension .NXP. This is a proprietary database file format provided with NEXPERT and has been satisfactory for the prototype development. This same format is also used to store intermediate results and to transfer data to and from external programs.

Data that may change over time is kept in a database. For example, all customer information is kept in a database that can be easily updated or changed as the customer base changes and does not require any change in the knowledge base. There is no actual reference to individual customers in the knowledge base.

When the information is read into the knowledge base from the database, each record of the database becomes a "dynamic object" in the knowledge base with each field becoming one of its properties. Dynamic objects are identified in the knowledge base by the "+" sign, which appears in the front of the object. For example, the list of customers is data that can and does change and so is kept in a database. The list of customers is read into the knowledge base and belongs to the NEXPERT class called "customer_class"; each customer becomes an object belonging to that class, which has several "properties" such as the type of gas, the capacity of the tank, and whether this customer requires delivery. The properties of these objects can be used in the rules for determining what actions will be taken on which objects.

Three external C programs are written to combine and format data and for a special interactive screen display. For example an NXP file is written which contains driver data and another which contains the date for which scheduling is to be done. An external program combines driver data with the list of trailers so that the drivers who are available for duty on the date that scheduling is to be arranged are combined with the trailers available. The final output of the system is a printout of delivery instructions for each driver for whom assignments have been made. The C program re-formats the data written by NEXPERT to a format similar to that actually in use at CLA.

To run an actual session of the Dispatcher's Assistant, the knowledge base is loaded and data on the drivers is retrieved. This includes such information as whether the driver is senior or non-senior, preference for type of gas, tractor assigned if the driver is a senior driver, and the schedule of drivers assigned to work the day, afternoon and night shifts for the week. The user has the opportunity to edit this data to update any changes that have occurred since the data was entered into the database or last updated. The day of the week and the shift for which scheduling will be performed is then selected.

The same situation occurs with the list of trailers, with the user having the opportunity, on screen, to edit the list of trailers to exclude any trailers that are temporarily unavailable, so they will not be assigned for a delivery. The list of customers is also read into the knowledge base in preparation for assignment. From this list of customers and based on the printout the dispatcher has from the existing prediction system, the user selects, from a multiple choice list displayed on the screen, all the customers who currently require delivery.

The expected delivery amounts are calculated by the system and a total delivery amount for each product within each zone is also calculated. The user selects which gas he wishes to concentrate on first for delivery and is then presented with the total amount of product for each zone. Next, the user selects which zone is to be assigned first for delivery.

5. **Methods**

Intermittent interview and observation sessions were conducted at the Canadian Liquid Air plant in Hamilton. An initial meeting was held before submitting a proposal to the Transportation Development Centre to assess the applicability of dispatching for the knowledge-based systems approach. After the award of contract, interviews were conducted as often as could be arranged. This was often very difficult because of the time constraints and pressures on the dispatcher and the developers experienced considerable difficulty in getting time with the expert.

Initial sessions were conducted as interviews, with the developers asking questions and the dispatcher answering them. The questions were asked in a general way to encourage free-flowing ideas and to get an over-all picture of the operation. This was necessary for the process of making a decision as to which area in the dispatching process to concentrate on for the prototype.

A couple of later sessions were spent observing the dispatcher while he performed his duties. When necessary, he would explain what he was doing, the procedures or regulations that he was following or taking into consideration and the reasons for decisions he might make.

These observation sessions, which included the opportunity for questions and answers, were very useful to the development team, and provided an opportunity to observe the kinds of pressures that the dispatcher is under as he conducts his duties.

All sessions with the expert were tape-recorded. Back at the development site, the development team listened to these recordings repeatedly. This proved to be a valuable tool when it was not possible to get as much time with the dispatchers as was desirable. At least the sessions that were conducted could be listened to again and again and as much information as possible extracted from each visit.

The guiding principles followed during the interview sessions were:

- allow total freedom of expression
- understand the dispatching operation
- gain insight into the potentially most effective user interface
- understand the pressures of the job
- understand the circumstances
- understand what is to be achieved
- understand the likely changes and ways of coping with those changes

6. User Interface

The completed prototype incorporates a multi-faceted user interface to demonstrate the flexibility and capabilities of today's technology.

The visual interface is designed as "screens" appearing on the terminal, which can either provide information to the user or retrieve information when necessary. The system is primarily menu driven for minimum cognitive load to the user. Information presented on a screen is relevant to the task at the time and the screens are designed so that the information required for that task is available and the user does not have to recall what had been entered on a previous screen.

Where appropriate, the screen interface has been designed with multiple choice selections where the user can select as many or as few choices as required. In other cases the data can be entered by typing or can be selected from a pop-up menu. Where data is entered by typing there is an opportunity to review the inputs and an opportunity to make any required corrections. Tables in which entries can be edited are typically used to provide the user with an overview before finalization. This provides an opportunity for changes and/or corrections. The use of colour and consistency of keystrokes influence the user's reaction to the system and they have been carefully controlled in the system. Screen displays of reports are available at different stages during the program and these reports can also be printed.

In addition to the normal keyboard-screen operation of the application, a hands-free interface based on a voice recognition system and a voice synthesizer has been integrated with the software tool. This provided a more flexible interface, which offers an attractive alternative to the keyboard in the eyes busy/hands busy dispatch environment.

The Dragon Systems, Inc. VoiceScribe system is used for the voice recognition. For this speaker-dependent recognition system, a limited vocabulary, specific to the expert system, was designed and the voice recognition was "trained" on it. It provides for voice commands equivalent to keyboard inputs, which also remain active throughout. Again, considerable attention was given to the consistency and logic of the commands. For example, the spoken word "Up" replaces the Up arrow on the keyboard, "Enter" spoken can be used instead of pressing the Enter key on the keyboard.

The questions appearing on the screen can also be posed with a voice synthesis system. The voice synthesis system selected was by AICOM Corporation and the synthesis is accomplished by sending text strings to the output device created by the voice driver. With the installation of the voice recognition and voice synthesis it frees the dispatcher from the computer terminal so that he can retrieve other documents and check on other information while still continuing with the program. These initial capabilities can be further developed and used all together or in any combination depending on the situation and the requirements.

As a further test of the feasibility of "hands-free" input, a voice activated text dictating system was provided. Since the dispatcher often needs to add free formatted text messages to the route delivery orders for the drivers it was thought to be convenient to input these notes also by voice.

The system selected for this purpose is DragonDictate from Dragon Systems, Inc. It is pseudo speaker-independent in so far as it starts with a general recognition of words contained in its extensive dictionary but presents a number of possible alternative interpretations. From the choices the user makes it refines its recognition for a particular speaker, thereby enhancing its effectiveness by developing a speaker-specific memory.

7. Results and Recommendations

The primary objectives of this project were to obtain experience in the application of expert systems to transportation-based scheduling and to provide a tool, usable by a dispatcher, that will mechanize a number of tasks, thereby lightening the presently overwhelmingly heavy workload.

After analysis of the tasks of the dispatcher, the functions and activities described above where selected for implementation because of their immediate effectiveness in supporting the work. The emphasis on the user interface throughout the project resulted in a reasonably effective interactive environment, which combines the traditional keyboard/screen interface with both voice input and output. With this prototype development, the team touched on the realm of multi-modal intelligent user interfaces and saw its potential in transportation applications.

Although the circumstances at the cooperating dispatch location did not make an actual trial use of the system practical, a demonstration given on location was very well received.

The prototype has been demonstrated to the dispatcher who was interviewed during the project, to his replacement who is currently doing the dispatching and to the head of the CLA office in Hamilton. Their reaction was highly positive and encouraging. They supported the development team and showed enthusiasm for continued development of the prototype.

After overcoming some initial difficulties with versions and releases of the NEXPERT system and after gaining a thorough understanding of the operation of the inference mechanism employed, the team found that a rule-based expert system is effective in implementing the required functions. In the case of the selected expert system, its capability to easily incorporate external modules written in a conventional programming language is beneficial.

A strong interest in the development has been demonstrated by the Canadian AI R&D community and has resulted in presentations at several conferences and workshops. AAI was also invited to demonstrate the system at a conference and exhibition in Detroit, sponsored by the American Association of Artificial Intelligence (AAAI89), the largest and premier AI conference in the world.

An off-shoot of the project is a product that was purchased by a major software corporation (Microsoft Corporation), even before active marketing. This product provides the voice hardware and software, plus software to interface to NEXPERT. The voice interface can be effective in many domains.

From the experience gained with this project, a strong recommendation can be made for the use of a capable expert system shell to implement similar functions.

For the particular application treated in this project it appears to be desirable to fully implement the specific capabilities and data required by an actual location in order to put the system to practical use and to achieve real assistance and workload relief. The next step would be to extract knowledge of the next higher level tasks from the appropriate domain expert and implement those, while improving the utility of the existing application.

Sommaire

1. Introduction

Au Canada, la vie des citoyens est largement conditionnée par les transports et activités connexes. L'immensité du pays fait que les distances à couvrir représentent souvent un facteur important à considérer. Et à mesure que le réseau des transports s'agrandit, sa gestion pose des problèmes de plus en plus aigus, en vertu de cette prémisse qu'il faut déplacer la plus grande quantité de marchandises et le plus grand nombre de personnes au moindre coût et avec un maximum de sécurité.

Au Canada, le transport routier de marchandises par des véhicules lourds est un impératif. Vers la fin de 1985 et au début de l'année suivante, le Centre de développement des transports (CDT) de Transports Canada et la firme Applied AI Systems (AAI) ont abordé le sujet des connaissances et du savoir-faire qui devraient être à la base des décisions en matière d'affectations de camions et de conducteurs. Les premières recherches ont pris la forme d'une rencontre avec la personne responsable des acheminements de gaz liquides à Air Liquide Canada Ltée, à Hamilton (Ontario), M. Dennis Ogilvie. Cette visite a permis de déterminer que le domaine des affectations se prête très bien à l'application d'une technologie fondée sur l'intelligence artificielle.

Comme tout régulateur, la personne responsable des affectations des conducteurs et des camions-citernes doit, compte tenu de tous les facteurs en jeu, pouvoir apprécier une situation donnée et en faire rapidement la synthèse. Une aide à la décision fondée sur une base de connaissances pourrait, en déchargeant le régulateur d'une partie des tâches qui lui incombent, lui permettre de se concentrer sur les aspects complexes, abstraits ou délicats de sa tâche qu'aucune machine, si intelligente qu'elle soit, ne saurait entreprendre dans l'état actuel des connaissances.

Le régulateur doit jour après jour apparier tracteurs, citernes, conducteurs et listes de clients, dans une ambiance d'astreinte perpétuelle où tout est en constante mutation, au point d'échapper parfois à tout contrôle. Il doit tenir compte des distances séparant chaque client et du temps de retour à l'usine, qui doit se faire à temps pour permettre le remplissage de la citerne en vue de la tournée suivante.

Il doit aussi faire face aux imprévus de toutes sortes, qui semblent surgir comme par exprès jour après jour : pannes mécaniques, maladie d'un conducteur, livraisons demandées avant le jour prévu, autant de situations qu'il faut régler sans que la marche normale du service ne soit perturbée.

En juin 1987, AAI avait remis à Approvisionnements et Services Canada une proposition spontanée sur le développement et la démonstration d'une aide à la décision fondée sur une base de connaissances et appliquée à l'affectation des véhicules routiers. Et en août 1987, AAI a obtenu le marché pour la mise au point d'un prototype du système.

L'objectif fixé à la recherche était de relever la sécurité et la productivité des véhicules et systèmes de transport routier de marchandises par le moyen d'une aide à la décision efficace en matière d'affectations et par l'introduction au Canada de la technologie des systèmes experts appliquée aux transports routiers. L'idée était que le prototype une fois mis au point pourrait aider les régulateurs ne bénéficiant pas d'une grande expérience à exercer un bon jugement dans l'exercice de leurs fonctions.

2. Domaine d'application

La personne responsable des affectations au service de distribution d'Air Liquide doit coordonner de façon optimale l'affectation des conducteurs, des véhicules et des destinations, de manière à assurer l'approvisionnement en temps voulu des clients. Exemple, un hôpital doit pouvoir compter sur des approvisionnements sûrs d'oxygène sans lesquels il ne peut fonctionner convenablement. Un client qui se trouverait à court d'un produit nécessaire subirait un préjudice financier énorme; des vies humaines pourraient être mises en danger. Bref, le régulateur doit arbitrer entre, d'une part, les aspects de rentabilité de son entreprise : exploitation à moindre coût et, d'autre part, les exigences du client qui peuvent fluctuer et parfois aussi à certains impératifs de sûreté. Le régulateur est soumis de la sorte à une pression considérable que viennent aggraver les nombreuses incertitudes et autres variables sur lesquelles il n'a souvent aucune emprise.

Vu le nombre de facteurs en jeu, et les mutations qui se manifestent jour après jour, si ce n'est d'heure en heure, on comprendra pourquoi la fonction régulation est si exigeante en temps et en compétence et pourquoi elle le devient d'autant plus qu'il y a plus de conducteurs, de tracteurs, de citernes, de familles de gaz à contrôler et de clients à approvisionner.

A côté des incertitudes et des mutations, certains impératifs invariables et permanents demeurent, comme la réglementation sur le transport des marchandises dangereuses qui stipule que la nature de la cargaison doit être affichée sur les citernes et que les formalités requises doivent être remplies par les conducteurs. Il y a aussi les restrictions sur les charges à l'essieu des poids lourds, imposées par les ministères provinciaux des Transports. Et les conventions collectives régissant les rapports entre la compagnie et les conducteurs qui doivent être respectées, comme par exemple la limite à six heures de conduite par période de 24 heures. Autant de facteurs limitatifs qui doivent être pris en compte.

A part ces facteurs, il en est d'autres qui, sans nécessairement relever de la réglementation, influencent la fonction régulation. Il peut s'agir de conditions imposées par les clients : sa cour est trop petite et ne peut recevoir des véhicules au-delà d'une certaine longueur. Les portes doivent être gardées fermées pour des raisons de sécurité et les livraisons ne peuvent se faire qu'à certaines heures précises. Le gaz liquide doit être livré à pression élevée et seules certaines citernes ont les pompes nécessaires. Ce sont donc de nouveaux facteurs qui s'ajoutent aux précédents.

Air Liquide Canada Ltée produit surtout des gaz liquides qu'elle distribue dans les régions du centre et de l'ouest de l'Ontario, dans le nord de cette province et à l'est jusqu'à Kingston. Ces régions sont divisées en zones de distribution. L'oxygène, le gaz carbonique et l'azote occupent sept zones et l'argon quatre. Les affectations sont fixées aux conducteurs à l'intérieur d'une zone donnée, l'idée étant que la citerne, partie pleine, doit revenir à Hamilton complètement vide.

Les quantités livrées sont mesurées avec précision et le conducteur doit remettre à son retour les reçus correspondants. Il faut près de trois heures tous les matins pour entrer ces données qui serviront par la suite à établir la liste des livraisons automatiques. Cette grille, fondée sur la consommation connue de chaque client, s'étale sur onze jours successifs. Le nom d'un client commence à y figurer quatre jours avant la date prévue pour la prochaine livraison, donnant au régulateur une certaine latitude dans sa programmation et lui permettant de regrouper au mieux les livraisons journalières. A partir de la grille produite par l'ordinateur, il établit la liste des livraisons par client et par famille de gaz de manière à pouvoir boucler les affectations.

Les clients ont le choix de la périodicité des livraisons, mais la plupart optent pour le système de livraisons automatiques. Il en est d'autres qui préfèrent appeler chaque fois que le réservoir descend à un certain niveau. Ce mode de livraison leur convient car ils ne savent jamais à l'avance combien ils peuvent consommer au jour le jour. D'autres optent pour des livraisons à date fixe, tous les lundis ou les jeudis, par exemple, ou tous les trois jours. La finalité d'un mode de livraison est que le client soit toujours suffisamment approvisionné.

3. Choix du système

La recherche a été organisée autour du système NEXPERT Object, un squelette de développement réalisé par la firme Neuron Data installée à Palo Alto en Californie. Bien que relativement nouveau sur le marché au début des travaux de recherche, cet outil présentait déjà une souplesse et une puissance manifestes. Le développement s'est fait sur un ordinateur compatible IBM 386.

NEXPERT est un système expert hybride à base de règles et orienté objets. Il comporte une puissante base de raisonnement qui lui permet d'inférer sur les connaissances ou les observations pour obtenir des affirmations ou déterminer des actions. Afin qu'il puisse s'intégrer aux ordinateurs existants, ce système a été conçu de manière à pouvoir fonctionner dans plusieurs autres langages et ainsi récupérer l'information et actualiser les bases de données. Il peut être implanté dans des programmes existants ou piloter des programmes externes en plus de gérer leur exécution.

Le système NEXPERT comporte un environnement de développement qui permet la rédaction de règles dans l'éditeur de règles sous forme graphique et dans un format apparenté à un chiffrier, et qui produit automatiquement et graphiquement la représentation de la base de connaissances et de sa structure de contrôle. Cette représentation visuelle du chaînage entre les concepts et les règles permet le déverminage des bases de connaissances et donne une image plus claire de leur structure. NEXPERT s'articule autour d'un noyau logique qui autorise les fonctions suivantes : chaînage avant et arrière, génération automatique de buts, multihéritages, filtrages et appels directs vers et depuis les programmes externes.

Pendant que se déroulait la recherche, le système était par ailleurs renforcé et perfectionné, tandis que de nouveaux produits faisaient leur apparition sur le marché. Les concepteurs ont eu de la difficulté à saisir tout le potentiel de cet outil. Par exemple, la version mode protégé du logiciel opératoire RUNTIME n'a pu être réalisée qu'après un effort considérable consacré à la résolution de problèmes qui, comme il fut découvert plus tard, étaient attribuables à la mémoire et qui ont pu être réglés par la mise à disposition d'une nouvelle version mode protégé du RUNTIME.

La version opératoire du logiciel RUNTIME est une interface qui diffère de la version développement. La fonction NORT (NEXPERT Objet Runtime) interprète les questions, les formules et les rapports rédigés en langage de définition d'écran, et que l'utilisateur peut entrer ou extraire. Un fichier de traitement séquentiel (.RTD) propose des idées, fait des suggestions, affiche le format pour l'entrée de données et, une fois ce processus terminé, affiche le format ou les fichiers pour l'édition des résultats.

L'une des raisons pour lesquelles le choix s'est porté sur le système NEXPERT Object a été l'importance donnée par les concepteurs à l'interface utilisateur. Or vu le caractère déterminant de cette interface dans le succès ou l'échec d'un système expert, il fallait s'assurer que le système choisi comporte ou puisse comporter une interface souple. Les modalités de ce choix sont expliquées en détail dans la section suivante. Le système NEXPERT offrait justement la possibilité d'avoir une interface utilisateur personnalisée pouvant être dotée de plusieurs des caractéristiques importantes aux yeux de l'utilisateur.

4. Description du prototype

Le prototype construit comporte une base de connaissances formée de deux fichiers, des données contenues dans des fichiers externes et qui peuvent être actualisées, et plusieurs programmes externes que le système peut exploiter.

Les deux fichiers formant la base de connaissances sont chargés consécutivement durant l'exécution. Ils contiennent l'ensemble des règles et les structures de base nécessaires au déroulement du programme. La structure de contrôle appelée aussi processeur de connaissances est un mécanisme qui, de façon intelligente et dynamique, lit et interprète les règles accumulées dans la base de connaissances. Celle-ci a été conçue de manière à pouvoir établir des chaînages entre les règles, ces chaînages servant à représenter la structure des connaissances accumulées et à gérer l'exécution du programme. Les règles sont écrites dans la forme SI "prémisses" ALORS "conclusion" dont voici un exemple :

```
If   there is no evidence of B10_no_trail_change
     And Execute "voice.exe"
     (@TYPE=EXE;@WAIT=TRUE;@STRING="6";)
     And there is evidence of trail_change
     And there is evidence of [|trailers|].avail
Then B20-change_trail
     is confirmed.
     And Create Object [|trailers|) |tr_avail|
     And TRUE is assigned to avail_trail
```

Les informations constituant la base de connaissances sont contenues dans des bases de données comportant une extension .NXP. Cette façon brevetée de structurer les bases de données a donné de bons résultats lors du développement du prototype. Ce même format sert également pour le stockage des résultats intermédiaires et pour le transfert de connaissances vers et depuis les programmes externes.

Les informations susceptibles de changer sont placées dans une base de données. La base de données clients, par exemple, peut être actualisée ou modifiée aisément sans avoir à intervenir dans la base de connaissances, qui, elle, ne contient aucune information de ce type.

Lorsqu'il y a transfert d'information de la base de données vers la base de connaissances, chaque item d'information devient un "objet dynamique" pour cette dernière et chacun des champs, une propriété particulière. Chacun des objets dynamiques dans la base de connaissances est précédé du signe "+". Par exemple, la liste des clients est un fichier dont le contenu est modifiable et doit par conséquent constituer une base de données séparée. Lorsque cette liste est transférée vers la base de connaissances, elle constitue au sein du système NEXPERT une classe particulière, la classe clients, et chacun des noms figurant sur cette liste devient un "objet" à l'intérieur de cette classe, caractérisé par plusieurs propriétés : famille de gaz, capacité du réservoir, périodicité des livraisons, etc. Ces propriétés font l'objet d'un traitement par la base de connaissances qui détermine alors ce qu'il y a lieu de faire à l'égard de tel ou tel objet.

Trois programmes externes écrits en langage C servent à combiner et à structurer les informations pour les afficher dans un format particulier sur écran interactif. Il existe par exemple un fichier NXP renseignant sur les conducteurs et un autre sur les dates devant figurer sur la grille des livraisons. Un programme externe met en correspondance le fichier conducteurs avec le fichier citernes de manière à apparier les conducteurs en service le jour prévu dans la grille des livraisons avec les citernes disponibles ce jour-là. Une sortie sur imprimante donne les instructions sur les livraisons que chacun des conducteurs désignés aura à faire. Enfin, le programme en langage C restructure les instructions produites par le système NEXPERT pour leur donner un forme semblable à celle en usage à Air Liquide.

Pour faire la démonstration de cette aide à la décision, on commence par charger la base de connaissances et par récupérer le fichier conducteurs pour savoir entre autres ce qui suit : ancienneté des conducteurs, famille de gaz qu'ils ont l'habitude de livrer, numéro du tracteur particulier qu'ils sont autorisés à conduire à titre exclusif en raison de leur ancienneté, et grille des conducteurs devant se présenter au travail le matin, l'après-midi ou le soir de chacune des journées de la semaine ainsi planifiée. L'utilisateur peut modifier cette grille pour tenir compte des changements survenus entre-temps ou depuis la dernière mise à jour. Enfin, le jour de la semaine et le quart devant figurer sur la grille des livraisons sont arrêtés.

Même chose à l'égard des informations sur les citernes, et que l'utilisateur peut, sur écran, modifier à son gré pour exclure, par exemple, celles qui sont temporairement hors service. Le fichier clients est lui aussi introduit dans la mémoire de la base de connaissances. Par recoupement entre ce fichier et la grille des livraisons automatiques, l'utilisateur détermine sur la liste à choix multiples affichée sur l'écran les clients devant être approvisionnés ce jour-là.

Le système calcule ensuite les quantités à livrer ainsi que le total par produit et par zone. Une fois que le choix de la famille de gaz à être livrée en premier est fait, le système en calcule la quantité totale par zone. Il reste enfin à déterminer la zone devant être approvisionnée en premier.

5. Méthodologie de la recherche

Des interviews et des séances d'observation ont eu lieu de façon intermittente à l'usine d'Air Liquide de Hamilton. Une première réunion a eu lieu avant que soit remise au CDT la proposition visant à évaluer l'applicabilité d'un système expert au domaine des acheminements. Une fois le marché attribué, des interviews ont été organisées chaque fois que possible, ce qui s'avéra peu souvent étant donné que le régulateur était très accaparé par son travail. En fait, les chercheurs ont eu beaucoup de difficulté à obtenir du régulateur qu'il les rencontre de temps à autre.

Les premières séances ont pris la forme d'interviews, au cours desquelles les questions posées étaient d'ordre général dans le but d'obtenir des réponses spontanées et de se faire une idée globale de la situation. Il le fallait afin de pouvoir isoler l'aspect particulier de la fonction de régulation devant servir à la construction du prototype.

Plus tard, deux séances ont permis d'observer le régulateur dans l'exercice de ses fonctions. Lorsqu'il le jugeait nécessaire, il commentait ce qu'il faisait et en expliquait la raison à la lumière d'une réglementation, d'une procédure ou autre.

Ces séances d'observation accompagnées de commentaires, très utiles aux yeux des chercheurs, ont permis à ces derniers de juger des contraintes qui pèsent ordinairement sur le régulateur.

Toutes ces séances ont été enregistrées pour que, de retour au bureau, les chercheurs puissent les écouter à loisir. C'est ce qu'ils ont fait, à plusieurs reprises. L'idée était d'autant meilleure qu'il leur a été difficile de rencontrer le régulateur aussi souvent qu'ils l'auraient souhaité et qu'ils ont pu ainsi tirer des séances qui ont effectivement eu lieu le maximum d'informations utiles.

Les principes qui ont guidé la conduite des interviews ont été les suivants:

- laisser toute latitude pour s'exprimer librement,
- pénétrer les fonctions du régulateur,
- pressentir l'interface utilisateur qui donnerait les meilleurs résultats,
- juger des contraintes qui s'exercent sur le régulateur,
- mettre ces dernières en contexte,

- saisir le but à atteindre,
- deviner les changements possibles et entrevoir les moyens d'y faire face.

6. Interface utilisateur

Le prototype achevé comporte une interface utilisateur polyvalente destinée à montrer la souplesse et le potentiel du système.

L'interface visuelle présente des "écrans" où s'affichent les informations voulues, ou qui indiquent quoi faire pour les récupérer. Afin d'alléger la charge de travail au maximum, l'utilisateur est guidé par des menus, tandis que seules les informations utiles à ce dernier sont affichées ou aisément affichables, le dernier message apparu sur un écran étant gardé en mémoire.

Pour plus de commodité, l'interface permet de faire des choix parmi plusieurs proposés, au gré de l'utilisateur. Il est également possible de dactylographier les données à entrer ou de les sélectionner grâce à un menu-éclair. Les données dactylographiées peuvent à tout moment être rappelées et corrigées, si besoin était. Des affichages tabulaires sur l'écran permettent de vérifier une dernière fois les données, de les modifier ou d'y apporter des corrections. L'emploi standardisé des couleurs et des commandes ayant une forte influence sur l'interactivité utilisateur-système, cet aspect a été soigneusement contrôlé. A divers moments durant le déroulement du programme, les résultats peuvent être affichés, voire imprimés.

Outre le dispositif d'entrée par clavier alphanumérique, une interface utilisateur-système de type "mains libres" a été prévue, comportant un système de reconnaissance vocale et un synthétiseur vocal. Cette approche confère au système plus de souplesse et remplace avantageusement le clavier, surtout dans un environnement aussi chargé que celui de la régulation des acheminements.

Le système de reconnaissance vocale choisi est le VoiceScribe de Dragon Systems Inc. De type monolocuteur, ce système dispose d'un vocabulaire restreint, spécifique à l'utilisation, pour lequel le système VoiceScribe a été entraîné. Les commandes générées correspondent à celles du clavier alphanumérique qui reste toujours actif. Là aussi, la cohérence et la logique des commandes ont fait l'objet d'une étude poussée. Par exemple, le mot parlé "Up" est l'équivalent de la flèche ↑ du clavier, et le mot parlé "Enter", de la touche "Enter".

Les questions affichées à l'écran peuvent également être communiquées au régulateur en mode vocal par l'intermédiaire d'un synthétiseur vocal, celui choisi étant fabriqué par AICOM Corporation. L'appareil est actionné par des suites de caractères transformées en signaux qui servent à la synthétisation du message parlé. Cet ensemble de reconnaissance/synthèse de la parole libère le régulateur du joug du clavier, et lui permet d'être mobile tandis qu'il donne ses instructions. Toutes ces fonctions encore au stade préliminaire peuvent être utilisées ensemble ou séparément selon la situation ou le besoin.

Pour vérifier plus avant la faisabilité d'une fonctionnalité "mains libres", un système de dictée à commande vocale a aussi été prévu. Il permet de produire les messages écrits non formatés qui doivent souvent accompagner les instructions données aux conducteurs.

Le système choisi en l'occurrence est le DragonDictate de Dragon Systems Inc. qui est un système pseudo-multilocuteurs, dans la mesure où, partant d'une reconnaissance globale des mots contenus dans son vaste vocabulaire, il propose un choix d'interprétations sémantiques. Par les choix que fait un locuteur particulier, le système finit par délimiter un champ sémantique, et par développer une mémoire, spécifiques à ce locuteur-là, ce qui ajoute à l'attractivité d'un tel système.

7. Résultats et recommandations

Les principaux objectifs de la recherche étaient d'étudier l'applicabilité d'un système expert à l'organisation des acheminements routiers et de mettre au point un outil permettant d'automatiser un certain nombre des fonctions assurées par un régulateur et, par là, d'alléger la tâche écrasante qui pèse sur lui.

L'analyse des tâches du régulateur a permis aux chercheurs de choisir parmi les fonctions et les tâches décrites plus haut, celles qui devraient être implantées dans un système expert, vu leur grande importance dans le déroulement efficace du travail. L'importance qu'ils ont accordé à l'interface utilisateur a débouché sur un environnement interactif raisonnablement efficace, du fait qu'elle conjugue des dispositifs d'entrée comme le clavier alphanumérique avec d'autres de type à commande et à sortie vocales. Avec le prototype qu'ils ont mis au point, les chercheurs ont abordé le domaine des interfaces multi-utilisateurs intelligentes, dont l'applicabilité à celui des transports leur est apparue pleine de promesses.

Bien que la conjoncture dans l'usine qui a collaboré à cette recherche n'ait pas été favorable à un essai formel du système, la démonstration qui y a été faite a donné de bons résultats.

En effet, la démonstration du prototype a été faite devant le régulateur qui avait été le contact de l'équipe de recherche, devant son remplaçant devenu entre-temps régulateur en titre et devant le chef d'Air Liquide à Hamilton. Leur réaction a été très encourageante et très positive. Ils ont appuyé l'équipe de recherche et manifesté un intérêt enthousiaste à l'égard du développement du prototype.

Ayant réussi à surmonter les difficultés des premières versions du système NEXPERT, et après s'être familiarisée avec le moteur d'inférences propre à ce système, l'équipe de recherche a trouvé que le système à base de règles choisi par elle se prêtait d'autant mieux à l'application envisagée, qu'il lui était facile d'intégrer des modules externes programmés dans un langage connu.

Le vif intérêt manifesté par les milieux canadiens concernés par la R&D en matière d'IA a débouché sur divers ateliers et conférences, tandis que la firme AAI était invitée à faire la démonstration de son système à l'occasion d'une exposition-conférence sur l'IA organisée par l'American Association of Artificial Intelligence (AAAI89), la première et la plus importante manifestation de son genre dans le monde.

Un important fabricant de logiciels, Microsoft Corporation, a acheté, avant même qu'elle ne soit lancée sur le marché, une variante de ce produit qui comprenait le matériel et le logiciel de commande vocale, ainsi que l'interface avec le système NEXPERT. L'interface vocale peut s'utiliser dans des systèmes appartenant à divers domaines.

L'expérience acquise grâce à ce projet de recherche permet de formuler avec force la conclusion que les fonctions dont il a été question ici se prêtent à un traitement automatique par système expert.

En ce qui concerne l'application particulière en question, il serait souhaitable d'intégrer au système toutes les fonctions et données propres à l'entreprise considérée, de manière à constituer une aide à la décision réellement utile et apte à alléger la tâche du régulateur. L'étape suivante serait de construire une base de raisonnement simulant celle qu'un expert dans ce même domaine utiliserait pour résoudre des problèmes à un niveau plus élevé, et de l'implémenter par le moyen de l'intelligence artificielle, tout en perfectionnant les connaissances acquises grâce au système actuel.

TABLE OF CONTENTS

	PAGE
1. Introduction	1
2. History of the Project	3
2.1 Initial Visit	3
2.2 Proposal	4
2.3 Area of Focus	5
2.4 Project Execution	6
3. Domain	8
3.1 Bulk Distribution Department	8
3.2 Drivers	10
3.3 Customers	11
3.4 Dispatching Process	12
4. Structure of the Expert System	13
4.1 Tool Selected	13
4.1.1 Description of the Tool	14
4.1.2 Developer's Interface	18
4.1.3 Runtime Interface	19
4.2 Overall Design considerations	20
4.2.1 Customer Needs	20
4.2.2 Cost-efficient Delivery	21
4.2.3 Driver Needs	21
4.2.4 Approach	22
4.3 System Organization	23
5. Structure of the Knowledge Base	27
5.1 Knowledge Base Files	27
5.2 Data Files	27
6. Principle of Operation of the Expert System	29
6.1 Knowledge Base	29
6.2 Computing Resources and Software	35
6.3 Visual and Keyboard Interface	36
6.4 Voice Interface	38

7. Operations Manual 40

7.1 Installing the System 40
7.2 Keyboard and Voice Commands 40
7.3 Starting Up A Session With Dispatcher's Assistant 41
7.4 Moving Within a Screen, and From Screen to Screen 43
7.5 DRIVER SCHEDULE Screen 43
7.6 Editing the DRIVER SCHEDULE Screen 44
7.7 SCHEDULING Screen 45
7.8 See Reports 46
7.9 TRAILER AVAILABILITY Screen 48
7.10 CHANGE AVAILABILITY Screen 49
7.11 TYPE OF GAS Screen 50
7.12 SELECT CUSTOMERS Screen 51
7.13 REQUIRED DELIVERY VOLUMES PER ZONE Screen 53
7.14 SELECT CUSTOMER/SELECT DRIVER Screen 54
7.15 Assign Trailers 57
7.16 FINAL REPORT Screen 59
7.17 SEE REPORTS Screen 60
7.18 END OF SESSION Screen 66
7.19 Dragon VoiceScribe 67
    7.19.1 Training VoiceScribe 67
    7.19.2 Using VoiceScribe 67
7.20 DragonDictate 68
    7.20.1 Bringing up DragonDictate 68
    7.20.2 Getting Started 68
    7.20.3 Train Vocabulary 68
    7.20.4 Using DragonDictate 69

8. Recommendations for Future Developments 72

8.1 Linkage with CLA's Prediction System 72
8.2 Importance of Voice-Interactive User Interface 72
8.3 Generalization of Methodology 73
8.4 Knowledge Engineering Requirements 73
8.5 Further Development 74

Appendix A: Listing of the Rules of the Knowledge Base A1

Appendix B: Configuration Files Required B1

Appendix C: Words Trained for Dispatcher's Assistant C1

Appendix D: Kepner-Tregoe Decision Analysis D1

LIST OF FIGURES

FIGURE	TITLE	PAGE
1	Customer Prediction Printout	9
2	Map of Customer Zones	10
3	NEXPERT Open AI Environment	16
4	Databases and Knowledge Bases	17
5	Rule Editor	18
6	Rule Network	19
7	Object Editor	30
8	Assigning Customers and Trailers	32
9	Driver Assignment	33
10	Example Multiple Choice Screen	37
11	Example Pop-up Screen	38
12	Opening Screen	42
13	DRIVER SCHEDULE Screen	44
14	DAY/SHIFT Scheduling	46
15	Interim Reports	47
16	Change Trailer Availability	48
17	List of Available Trailers	49
18	Type of Gas	50
19	Select Customers	51
20	Delivery Volume Per Zone	53
21	Customers and Equipment	54
22	Assigned Customers	56
23	Assigning Drivers to Trailers	57
24	Assignment Summary	59
25	List of Reports Available	61
26	FINAL REPORT	62
27	SELECTED DRIVERS	63
28	Trailers for Particular Gas	63
29	AVAILABLE TRAILERS	64
30	CUSTOMERS	65
31	END OF SESSION	66
32	DragonDictate Window	68
33	Using Dictate Microphone	70
34	Selecting Correct Word	70

1. Introduction

The "Knowledge-Based Dispatcher's Assistant" project was funded through the Transportation Development Centre of Transport Canada and the Department of Supply and Services, Canada as part of the Unsolicited Proposals Program. The goal of the project was to introduce expert system technology to surface technology systems in Canada and, in particular, to provide the means for a consistent and improved level of decision making in the dispatching process and to raise the level of efficiency of the dispatcher, thus allowing him to direct his attention to tasks beyond the range of the expert system.

The human dispatcher is required to reason upon a multiplicity of criteria, make quick judgments and integrate the results. An advisory knowledge-based system that assisted the dispatcher with some of the tasks would then allow him/her to concentrate on more complex, abstract and challenging tasks which are beyond the capability of existing computer-system technology.

Canadian Liquid Air Limited (CLA) agreed to have the knowledge engineers observe and question their dispatcher as he performed his duties at his office in Hamilton, Ontario. His duties included assigning drivers with the appropriate tractor and trailer to deliver the product to the customer.

The dispatcher is constantly under pressure to complete his daily assignments of tractor, trailer and list of customers for each driver. The afternoon shift of drivers begin their duty around 15:00 and if the assignment is not complete, they wait in the coffee room, on company time, until the assignments are complete. In making the assignments, the dispatcher must take many factors into consideration, including the distance between customers and the distance from the plant. The trailer must be returned in sufficient time for it to be refilled with product and ready for the next shift of drivers. In the midst of this, the dispatcher must cope with the unexpected emergencies that seem to arise each day. A breakdown of equipment, a driver phoning in sick, or a customer needing a delivery sooner than expected can happen on any given day. These situations must be dealt with while still providing the service to the customer.

The process of developing a knowledge-based system involves spending time with those who are experienced and knowledgeable in the domain. It means learning their terminology, studying and observing their approach to the problem in question, and developing a relationship so that they are comfortable in describing the thought processes they use in performing their tasks. This acquisition of knowledge and understanding of the domain is time-consuming but very important for the implementation of the knowledge-based system.

From the beginning of this study, unforeseen difficulties limited our time with the expert and this problem persisted throughout the project. Due to the difficulty in obtaining sufficient time to go through an iterative process of development and demonstration, there was a shifting of emphasis in the project to a concentration on developing an effective, flexible, and advanced user interface. Since one of the main determining factors in the success or failure of expert systems is related to the user interface this is an important aspect in the development.

Attention was given to the development of the screens that a user would see while using the system. This included emphasis on colours, placement of windows and menus and choice of keys to be used in activating the system. In a traditional interface the interaction is through the use of the keyboard. In this project the use of voice was introduced as an alternative interface. Both voice recognition and voice synthesis was used in the project. Since transportation applications are traditionally "hands busy/eyes busy", the use of voice gives an attractive alternative to the interface with the computer.

2. History of the Project

Canadian Liquid Air Limited (CLA) distributes liquid gases from its plant in Hamilton, Ontario to its broad customer base in central, western and northern Ontario and as far east as Kingston, Ontario. The other more easterly parts of Ontario are serviced from the Montreal office. There are two parts to the operation in Hamilton - the production plant where the liquid gas is produced and the Bulk Distribution Plant from where the product is dispatched to the customers by means of bulk carriers.

The customers range from steel mills to food processing companies, and from hospitals to automobile manufacturers. The requirements for the liquid gases are thus quite varied. Irrespective of the business and why the customer uses the product the mandate of CLA is to provide each customer with a constant supply of the liquid gas. Any interruption in supply to the customers can cause serious delays in their production and could be very costly for the customer. In the case of a hospital, it could be life-threatening.

On the recommendation of Mr. Russ Barker, Manager, Gas Distribution, Montreal, and Mr. Lewis Sabounghi, Scientific Authority for the project from TDC, the plant in Hamilton was selected for this study.

2.1 Initial Visit

In January 1986, an initial visit was made to CLA in Hamilton and an interview was conducted with Mr. Dennis Ogilvie, the senior dispatcher for that office. As the senior dispatcher he is responsible for assigning each of his drivers on each shift to deliver the required gas to selected customers as well as for assigning which tractor and which trailer they will use to complete their assignments.

To complete this task in the most cost-effective manner while still maintaining customer satisfaction is very important. To meet this mandate many factors must be taken into consideration, not the least of which is the individual drivers who will be delivering the product. Certain union regulations must be observed: for example, the senior drivers only work day shifts. The dispatcher must constantly be aware of potential jealousies among the drivers if one driver feels another is getting more of the preferred routes (all drivers prefer runs where they have a long drive and deliver their full load to one customer).

Factors around the individual customers are also taken into consideration. For security reasons, some customers lock a gate after a certain hour. This is the entrance to their yard for unloading of the product so this customers delivery must be scheduled during the hours that his yard is open to accept delivery. Some customers have erratic and unexpected surges in the amount of product they require and the dispatcher has to be able to respond to their needs, often with very short notice.

2.2 Proposal

Following the initial visit, and based on the observations and information obtained at that visit, it was felt that a knowledge-based system approach could be developed in the dispatching domain which could be of benefit to the dispatcher in performing his daily duties.

It was observed that there were very heavy demands on the dispatcher's time and that he was constantly under pressure. The afternoon shift of drivers report for duty at 15:00 and the dispatcher must have the assignments for those drivers complete, otherwise valuable time would be wasted while this was being completed. The assignment for the day shift for that day had been completed the afternoon before. In assigning the customers to the day shift he has to take into consideration the amount of time driving to each of the customer sites, the amount of time to unload the product and return to the plant at Hamilton and ensure that there is sufficient time for the truck to be refueled and loaded with product just in time for the afternoon shift driver to set out on his assignment. This is just one example of the pressure and constraints that the dispatcher is facing each day. Any system which could relieve some of these pressures or take over some of the tasks would be of benefit to the dispatcher. He would then have more time to devote to other higher level activities which are more creative or require more human input.

Applied AI Systems, Inc (AAI) sent a draft proposal to Mr. Lewis Sabounghi at the Transportation Development Centre (TDC) in Montreal and in June 1987, AAI submitted a proposal to Department of Supply and Services. A contract for "Development and Demonstration of Knowledge-Based Vehicle Dispatcher's Assistant" was awarded in August of that year.

2.3 Area of Focus

The knowledge engineering process involves repeated sessions with the expert(s) while he (she) describes what tasks he performs, the process of how he performs these tasks, and the factors which influence his decisions. This knowledge acquisition process usually involves frequent and intense interviews with the expert(s) which generally become more structured as the knowledge engineer gains an understanding of the process and can focus in on areas which are still unclear and ask more specific and direct questions. After each knowledge engineering session, the information as it is understood should be interpreted into rules and implemented in the prototype. As part of the prototype development process the expert should review the prototype, comment and make suggestions on what has been done and how it should be changed. This iterative process needs to be repeated many times until the expert's know-how has been crystallized in the system. Throughout this iterative process the user interface is refined and changed until it is functional and suitable for the domain in which it will be used.

From the beginning, it was difficult to obtain time with Dennis Ogilvie, the "expert" in Hamilton. Due to labour problems, permission for the first visit was not received until November 1987 even though the contract had been awarded in August of that year.

At that meeting in November several possible application areas were discussed and analyzed. The possible areas under consideration included:

1) Equipment Management
2) Schedule Generation Advisor
3) Schedule Alteration Advisor
4) Vehicle Operation Advisor
5) Legal Compliance Advisor
6) Driver Scheduling Advisor
7) Maintenance Scheduling Advisor

The area selected for development of the knowledge-based system was "Driver Scheduling Advisor". A copy of the Kepner-Tregoe Decision Analysis method that was used to determine the best application area is included in Appendix D.

2.4 Project Execution

Once the application area was established another visit to Hamilton was required to concentrate the efforts on extracting the knowledge as it applied to that area. Again there was considerable difficulty in getting a time when an appointment could be made. It was postponed a couple of times until a visit was arranged for the 11-14 January, 1988. These delays in obtaining time with the expert hindered the progress of the project.

In mid-February 1988 arrangements were made for another visit to Hamilton in the first week of March. When the time approached, the meeting was postponed. There were repeated phone calls but his work load was always very heavy and it was not possible to arrange another visit until June.

Work continued on the prototype but it was more difficult without the expert's feedback and comments. The prototype was developed with little input from the expert as to the user interface or the correctness of the algorithm. Each time a visit was made to CLA in Hamilton, the sessions with the expert were recorded. These tapes were listened to as many as five or six times in an attempt to extract as much information as possible and verify that what was being implemented was accurate. Building knowledge-based systems is very labour intensive and without ready access to the expert and feedback from him the task became very difficult. This was one of the primary reasons for a shift in the emphasis of the project to the user interface.

Following a careful evaluation of the available tools for the implementation of the knowledge-based system, NEXPERT Object was selected as the tool of choice. An amendment to the contract was required because this tool was more expensive than previous tools which were being considered. One of the reasons for the selection of NEXPERT Object as the best tool for the task was the facility to develop a separate runtime interface so that the development and runtime environment are completely different. Again, this is an important consideration in development since the user interface is so important. The Protected Mode Runtime operates above the 640K barrier and so with the installation of this new runtime some of the memory problems which were experienced early in the development were eliminated.

There were some difficulties with the tool, as described later, but these were overcome and the decision to choose NEXPERT was not regretted. Its combined object-oriented and rule-based capabilities were important features in the development. NEXPERT Object was purchased in January 1988 and a 386 IBM compatible computer was purchased for development in March of the same year. In the early months of 1989 an updated version of the software development package was received as well as a Protected Mode Runtime system. The update version of the development system added some capabilities which had been required for the prototype and resulted in some restructuring of the knowledge base.

In January 1988 the contract was amended to include the addition of a voice-driven interface to the prototype. The objectives in adding the voice interface was to explore and demonstrate the capabilities of voice recognition/synthesis technology and to add robustness and flexibility to the user interface. Transportation applications have been recognized as an area where voice technology is very effective and sometimes the most efficient means of communicating with the computer. Many transportation applications are described as "hands busy/eyes busy". The use of voice in these instances is a very important alternative to the keyboard as an interface to the computer.

The DragonScribe VS-1000 User System to allow for voice commands of the program was purchased, installed, and implemented and the Aicom text-to-speech synthesizer was purchased, installed and implemented.

Following discussions which were initiated in April 1989, in January 1990 the contract was amended to include the purchase of DragonDictate. The dispatcher often wants to give the driver special instructions such as instructions as to which customer to fill first, directions to the location of a new customer, or what to do if he encounters a potential problem at a certain customer site. These instructions can be dictated to the system and printed.

3. Domain

Canadian Liquid Air Ltd (CLA) in Hamilton, Ontario agreed that Applied AI Systems, Inc could interview their dispatcher regarding the process of dispatching his drivers to deliver product to their customers.

3.1 Bulk Distribution Department

The department within Canadian Liquid Air Limited which dispatches the drivers to deliver product to their more than 450 customers is the Bulk Distribution Department. The majority of the product that is delivered is in liquid form - liquid Oxygen, Nitrogen, Carbon Dioxide and Argon - but Hydrogen is delivered in gas form. The Sales Department is responsible for signing up new customers and providing the Bulk Distribution Department with the necessary information about the customer. This information tells the dispatcher which product the customer will be receiving, the hours that his yard is open for delivery, how his delivery is to be scheduled, the capacity of his tank, and some information about the nature of his business and how the product will be used.

The Bulk Distribution Department is then responsible for ensuring delivery to the customer while taking into account all the factors which must be considered in making the delivery. The customer and driver consideration are described below. In addition there are many government regulations which must be followed. For example, the Bulk Distribution Department is responsible for ensuring that all the proper licenses and permits are obtained from the Department of Transport. The weight restrictions for each area where trucks are dispatched must be known and obeyed. Since they sometimes have to go outside of the province and into the U.S. their weight restrictions must be known and permits must also be obtained to travel in these areas. The regulations set out by the Transportation of Dangerous Goods Act must be followed. This includes ensuring that each driver is given the minimum required training of fours weeks in the characteristics of the products they are transporting under various conditions; ensuring that the proper paperwork has been completed by the drivers and that it is recorded and filed; ensuring that the driver has the correct documentation with him at all times which names the product and the amount of product which he is transporting; ensuring that the trailer is equipped with the required placarding located in the specified location, and with the required size and description; and ensuring that the required UN # is in place on the vehicle.

When the customer receives product, a meter on the trailer records the exact amount of product that is delivered. This record is given to the Bulk Distribution Department and approximately three hours is spent each morning inputting this record of the work done on the previous day. This data is used by the existing computerized prediction system for predicting when the customer will require delivery based on the past consumption. After the data has been entered and processed, then a printout is posted on the bulletin board of the office. See Figure 1. The printout is for customer needs for eleven days and a customer's name starts to appear on the printout four days before it is actually critical that his delivery take place. From this printout, the dispatcher makes a list of customers for each gas which require delivery and he is going to assign.

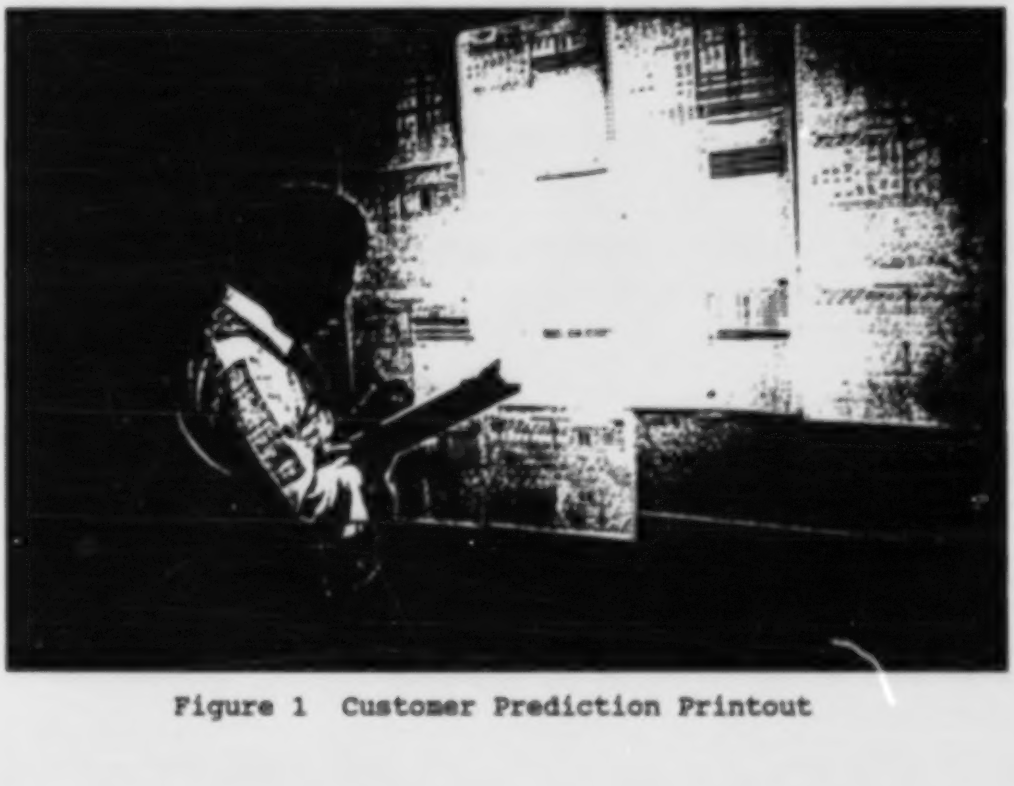


Figure 1 Customer Prediction Printout

Customers are divided into zones and in planning the delivery, any one driver does not cross zones. That is, all the customers he delivers to on one run are within the same zone. A large map on the office wall shows the division of zones and this is referred to often by the dispatcher. See Figure 2.

Figure 2 Map of Customer Zones

3.2 Drivers

Drivers who deliver liquid gases must be the very best drivers because of the dangerous goods they are transporting. They must have a minimum of three years experience driving other tractor-trailers before they can be considered for the delivery of the liquid gases. To deliver the product to the customers CLA initially had two shifts, which was later increased to three. There are nineteen bulk carrier drivers delivering product and of these nineteen, seven are senior drivers. There are certain benefits and privileges which the seven senior drivers enjoy. These benefits include working only the day shift, delivering their choice of gas to the customers and in most cases they operate the same tractor.

There are union and government regulations about the drivers which must be observed. For example, a driver can only drive a maximum of 6 hours in any day. In planning the routes and schedules this must be given careful consideration. For lengthy runs, the dispatcher wants to avoid having the driver required to stop for a rest if only another hour would complete the run. On the other hand, when overtime is required, it must be fairly distributed among the drivers and shift premiums must be observed.

3.3 Customers

The schedule for when delivery should take place to customers is negotiated with each customer. The options include:

Call In - A customer phones the dispatch office when he requires delivery. It is the responsibility of the customer to observe the level of his tank to ensure that he does not run out of product. Once the customer calls the dispatcher and a commitment is made as to when the delivery will take place this will not be changed.

Set Number of Days - Delivery to the customer is at a regular interval as required by the customer. For example, if he specifies that he wants delivery every three days, then on every third day product will be dispatched to his site and this does not change. Once this is established, the dispatcher does not alter this schedule.

Systematic - Product is delivered on a fixed schedule every week and this does not change. The customer may specify that he wants delivery every Monday and Thursday and that is the basis on which the deliveries are made. Again, this is established with the customer and the dispatcher will not alter the schedule. When the customer's specified day comes, he will receive product.

Forecasted - The majority of the customers follow this schedule type. An existing computerized system predicts when the customer requires delivery based on the history of prior consumption. A computer printout is produced each day which lists the customers requiring delivery with a lead time of four days. That is, the customer's name starts to appear on the printout four days before it is critical that he receive delivery. This gives the dispatcher some flexibility in determining his delivery schedule and route depending on the number of drivers available and the location of the customers.

3.4 Dispatching Process

When the dispatcher is ready to do his assignments he goes to the computer printout of customers which is posted on the wall. With his clipboard in hand, he write each gas at the top of the page and then lists the customers for each gas which require delivery. On return to his desk he has a listing of which drivers are on duty for each shift and the tractors and trailers available. The assignment is completed for the afternoon shift first since it must be ready for those drivers when they arrive for work. The decision of which customers to deliver to in the afternoon is determined by many factors. Most important is the urgency of the customer for product and any prior commitment or promise that has been made. If a call-in customer has requested product for that day and it has been promised, then that will not change. The dispatcher must also remember the times when he can deliver to the customer as some customers lock the gates and there is no access after business hours.

In grouping the customers together for delivery, the dispatcher keeps in mind that he should not cross zones. He must also take into consideration the amount of product the trailer can carry, the total amount required by the customers assigned, the time to travel to each customer and the time to pump off the product at each customer site.

In assigning the trailers to the delivery, the dispatcher needs to know the pressure in each customer's tank. A trailer must be assigned which can pump the product above the pressure in the customer's tank. Certain trailers are always assigned to certain customers. For example, Trailer 414 always goes to General Motors unless it is going north. In doing the assignment, the largest rigs are always assigned first.

4. Structure of the Expert System

4.1 Tool Selected

The "NEXPERT Object" expert system development tool, from Neuron Data of Palo Alto, California, was selected as the tool for the implementation of the software prototype "Dispatcher's Assistant".

NEXPERT Object was selected as the best tool for this project after a careful evaluation of the leading expert system development tools. Its graphical interface, powerful reasoning mechanism, open architecture and object-oriented approach were some of the features that led to the decision to select NEXPERT. The advantages of NEXPERT were not, however, immediate. At the beginning of the project NEXPERT was still a young product and during the course of the project there were upgrades, refinements and even a new product (NEXPERT Protected Mode Runtime) introduced. These changes sometimes required a restructuring of the knowledge base and often a new release was required to solve a problem that the development team was facing. Despite the initial inconsistencies and problems we feel that so many important issues were addressed by NEXPERT that its benefits outweighed the problems.

There is always an element of uncertainty when working with new and rapidly developing software. One is sometimes not sure whether unexpected behaviour by the system is due to insufficient understanding on the part of the designer, due to a software bug or even due to a shortcoming of the computer system.

Resolving these difficulties involved a lengthy, but necessary learning process. As an example, it can be mentioned that a series of screens developed in an earlier phase of the development were made useless by an update of the software. Another important capability, the ability to show the data retrieved from a database file in a table format which the user can edit, which was necessary and greatly influenced the development, was not documented and became known only through an incidental conversation when one of the team visited the Neuron Data headquarters in California.

Such findings do not diminish the impressive potential of the NEXPERT tool, the potential of which is evidenced by its rapid acceptance by industry at large. But rather this is a common experience when working at the leading edge of technology.

4.1.1 Description of the Tool

NEXPERT Object is a hybrid rule-based and object-based expert system building tool. A rule is written in the format of

if...then do...

and is the mode for representing dynamic chunks of knowledge and for linking facts or observations to assertions or actions. The objects are high level structures used to describe complex elements of knowledge and their structural organization. Objects can be dynamically created by the rules, as is the case in this prototype, during the execution of a session. These dynamic objects belong to classes and therefore can inherit properties and their values. The rules can address these objects via their pattern-matching functionalities.

NEXPERT Object is built around a powerful reasoning kernel, which integrates the following capabilities:

(1) Forward and backward chaining;
(2) Automatic goal generation;
(3) Multiple inheritance;
(4) Pattern matching mechanisms; and,
(5) Direct calls to and from external routines.

(1) Forward and backward chaining

Many problems are presented as sets of data from which conclusions must be drawn (forward chaining). NEXPERT allows this reasoning from facts. The goals which are generated in this forward chaining process can be ordered, and in doing so, the system's "focus of attention" is controlled. NEXPERT can also reason from goals (backward chaining) and navigates through the knowledge base, discovering the rules and subgoals relevant to the user's and/or the system's goal(s).

(2) Automatic goal generation

In evaluating a problem, a system needs to automatically generate alternative goals. These alternative goals are automatically generated as the factual knowledge is matched to the problem. NEXPERT ensures the coherence of the resulting focus of attention, and allows one to control the consequent evaluation of those alternative goals.

(3) Multiple inheritance

The hierarchical relations between objects allows one to pass properties and their values from one object (parent) to another (child). An object can inherit the value of any of its properties from a parent object and it can also have two or more parents. There is also a user defined inheritance strategy which allows customization of the way any given property of an object should inherit values from higher structures and this is defined in a meta-slot.

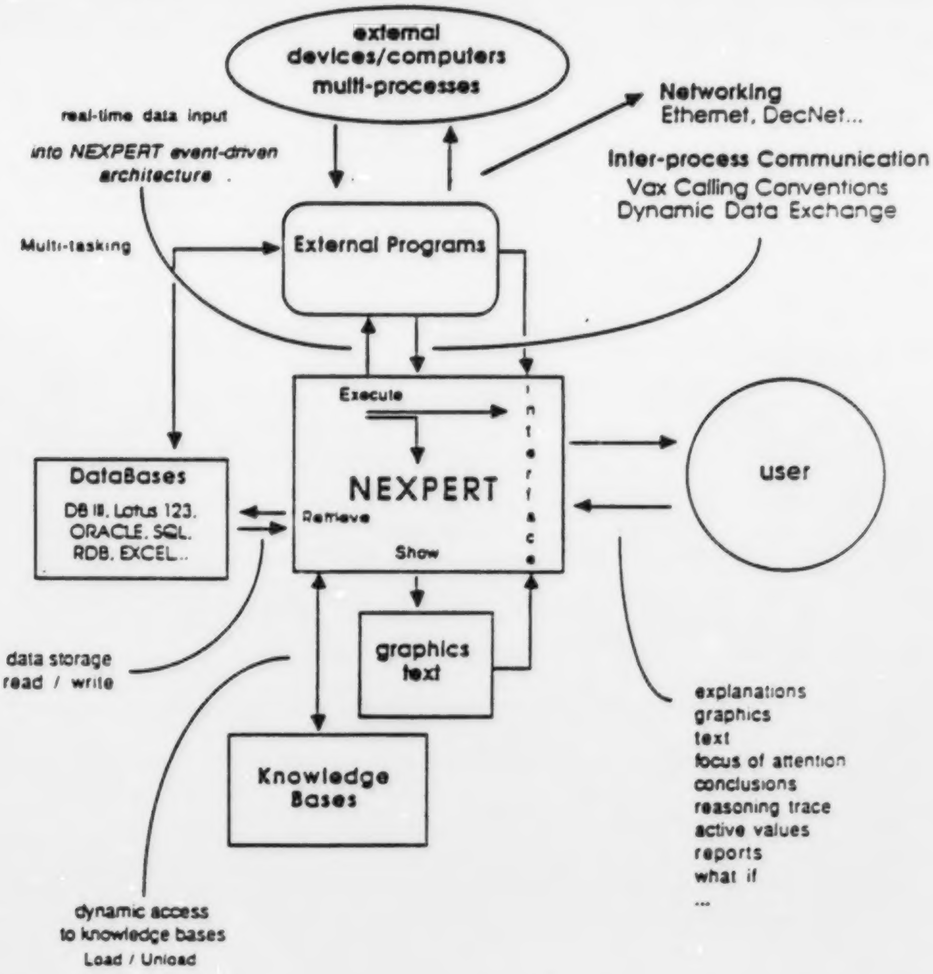
(4) Pattern matching mechanisms

Through the object-based description of the domain knowledge in NEXPERT, the inference engine has the ability to reason about facts involving many objects, called patterns. Operations on the left hand side of a rule (the "if" side) creates a list of objects which satisfy the specified criteria, and this list can then be re-used in the right hand side of the rules and actions can be taken on that list.

(5) Direct calls to and from external routines

NEXPERT can call user-written procedures and pass parameters to these procedures or NEXPERT can be controlled by an external program, as shown in Figure 3. External procedures are called by "Execute" statements from within NEXPERT and are used in the prototype developed for TDC. The Execute statements can be placed in either the conditions (LHS) or actions (RHS) of rules. When the inference engine processes an "Execute" statement, it calls the user written procedure which has been associated with that "Execute" statement.

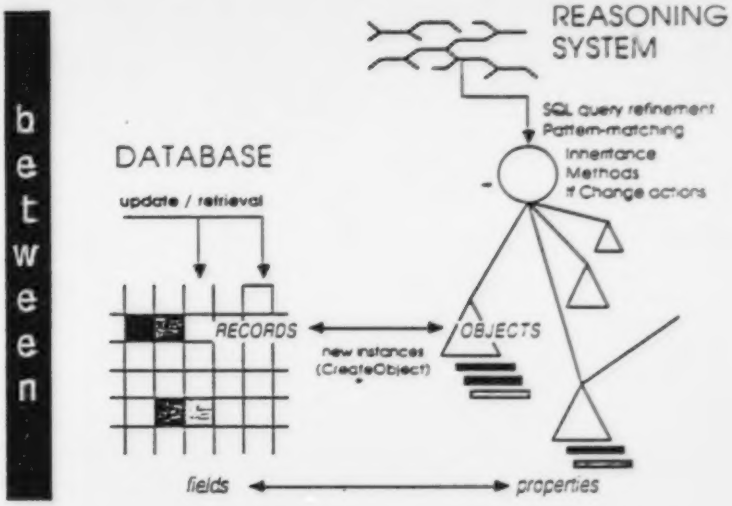
Database links allow NEXPERT to access data stored in external files or databases. This allows a clean separation of knowledge and facts. Knowledge is stored in knowledge bases, and facts in external files or databases. The database links allow the application to retrieve data from the database, Figure 4, and to write the results of its reasoning back into the database, or to create a new data file. Database records are associated with NEXPERT objects, database fields are associated with NEXPERT properties, and database cells are associated with NEXPERT object slots.


external devices/computers multi-processes
real-time data input
into NEXPERT event-driven architecture
Networking
Ethernet, DecNet...
Inter-process Communication
Vax Calling Conventions
Dynamic Data Exchange
Multi-tasking
External Programs
Execute
interface
NEXPERT
Retrieve
Show
DataBases
DB III, Lotus 123, ORACLE, SQL, RDB, EXCEL...
user
graphics text
data storage read / write
Knowledge Bases
explanations
graphics
text
focus of attention
conclusions
reasoning trace
active values
reports
what if
...
dynamic access to knowledge bases
Load / Unload

Copyright © 1987 by Neuron Data Inc. All rights reserved.

Figure 3 NEXPERT Open AI Environment

NEXPERT is integrated with standard relational databases and spreadsheets. One level of integration is through transparent, built-in bridges to both spreadsheet formats and other specified databases, and the other level of integration uses separate bridges to connect NEXPERT to the major relational databases on the market. The built-in bridge for the NXP format was used in this project to retrieve data.


between
REASONING SYSTEM
SQL query refinement
Pattern-matching
Inheritance
Methods
If Change actions
DATABASE
update / retrieval
RECORDS
new instances
(CreateObject)
OBJECTS
fields
properties

Databases & Knowledge Bases

Entity type		Class
Entity instance	⟷	Object
Attribute	⟷	Property
Attribute value		Property value

Copyright © 1987 by Neuron Data Inc. - text and graphics

Figure 4 Databases and Knowledge Bases

4.1.2 Developer's Interface

The NEXPERT Object development system provides a window based user interface for the developer. The rules are displayed and edited in a graphic, spreadsheet-like RULE EDITOR. See Figure 5. The NETWORK tool automatically creates a graphic representation of the knowledge base and its control structure. This provides a visual representation of the links between the concepts and the rules and is a browsing tool which permits both deductive and evocative navigations in the knowledge base. See Figure 6. There is also a network which displays the Object structure and the relation between objects, subobjects and classes. This allows the knowledge designer to follow the inheritance relations.

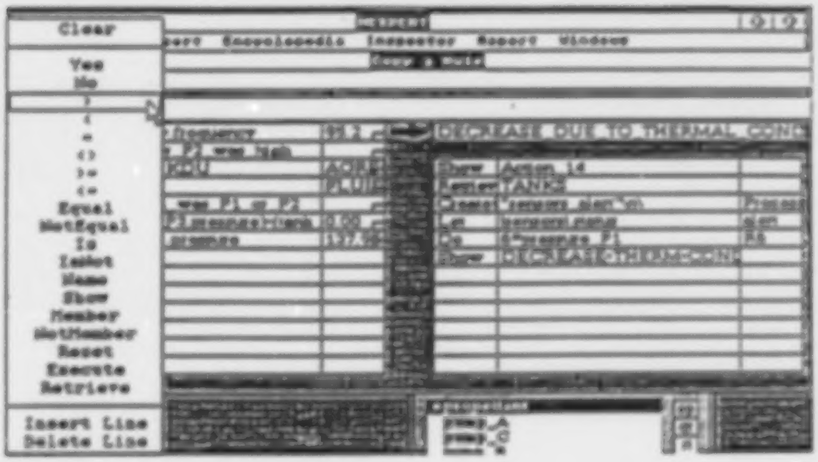

NEXPERT
Clear
Encyclopedia
Inspector
Report
Windows
Yes
No
?
<
=
<>
>=
<=
Equal
NotEqual
Is
IsNot
Name
Show
Member
NotMember
Reset
Execute
Retrieve
Insert Line
Delete Line
DECREASE DUE TO THERMAL_COND
TANKS
pump_A
pump_C

Figure 5 Rule Editor

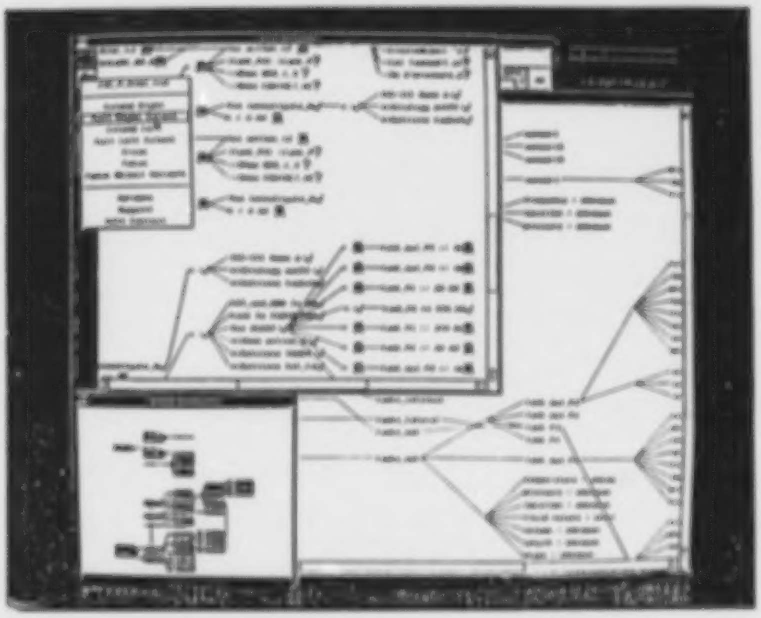


Figure 6 Rule Network

4.1.3 Runtime Interface

The Runtime delivery system is a different interface from the development version. NEXPERT Object Runtime (NORT) is a character-based interface program for executing NEXPERT applications. NORT interprets questions, forms, and reports written in the NORT Screen Definition Language for user input and output. Screens can be built using the Screen Painter program (which comes with the Runtime) or any text editor. The developer specifies a control structure to the knowledge-based application, as well as an end-user interface. The control structure is a batch file which volunteers data, suggests hypotheses, displays data entry forms, and shows report forms or files once it is finished. Among the capabilities of NORT's interface are: tables, lists, pop-up menus, edit lines, and text boxes.

4.2 Overall Design Considerations

The domain of dispatching as an application area for an expert system was chosen because of its complexity and the importance of the task. Actual experience by the cooperating company has shown that it is difficult to train a person for this work. When the initial preliminary visit was made to the Hamilton site, the senior dispatcher, Mr. Dennis Ogilvie, was in the process of training a replacement to take over his dispatching duties. After six months the new dispatcher was not trained and he had left the position.

This attests to the complexity of the task of dispatching and also to the amount of stress associated with the position. On the one hand, the dispatcher must meet his customers needs and make certain that the customer never runs out of product. On another hand he must keep the costs down as much as possible and deliver the product in the most cost efficient manner. The dispatcher's task is further complicated by the importance of maintaining a good rapport with the drivers and avoiding creation of any jealousy among the drivers.

4.2.1 Customer Needs

Understanding and quantifying the customer's needs is the first step in the dispatching process. The computerized system already in place at Hamilton gives a four day warning of when a customer is going to need product to be delivered (for customers which are on the forecasted delivery schedule). From this information, the dispatcher can judge how much flexibility he has in supplying a customer without significant danger that any particular customer will run out. For some customers with whom he may have a good relationship, he will give them a call to verify if they in fact need the product to be delivered on that day or can wait for one more day. In reality, when making assignments, the dispatcher is juggling several days and shifts simultaneously. He is always thinking one or two days in advance. For example, if a customer who is supposed to receive a delivery today can wait until tomorrow, he will have a truck in that area and the two deliveries can be combined for a run that will use all the product in the trailer and thus give an efficient run.

There are many factors about a customer which the dispatcher must keep in mind while dispatching. For example, he needs to know whether the customer operates on a twenty-four hour basis. If he is using product twenty-four hours a day, the span of time during which his delivery can be delayed is probably much less. He needs to know certain information about the customer site. Not all customers can handle the big rigs in their yards and this must be taken into consideration in doing the assignments. Some customers lock their gates at 18:00 and so delivery by the night shift is not possible. Some customers have peak seasons when their volume may actually double and the dispatcher needs to be able to anticipate this change and prepare for it.

4.2.2 Cost-efficient Delivery

The overriding rule is to send out a rig with a full load and have it return to the plant empty. The dispatcher juggles which customers he assigns to be delivered to on which day and by which capacity of trailer to make the most cost-efficient delivery. Financial reports are produced which report on the volume of each product which is delivered over a specified period of time. These reports also calculate the cost per volume of liquid delivered. This cost is often a direct reflection of the skill that the dispatcher has in doing his assignments. If routes and customers are skillfully organized and assigned this will be reflected in the financial reports. But, this must be achieved while still maintaining customer satisfaction.

4.2.3 Driver Needs

While keeping the needs of the customer in mind and the most cost-efficient means of meeting their needs, the dispatcher must also consider the drivers who will be delivering the product. If there is labour unrest and a strike results, this will cost the company money. A new driver requires about 3 months training in the handling of the gases. There are dangers involved in the handling of the gases and it is important that the drivers be properly trained in procedures in Handling the gases and to know what to do in an emergency. Driver satisfaction is a very important aspect to the delivery of the gases and once the investment in training has been made, the company naturally tries to keep their drivers. It is the responsibility of the dispatcher to see that the union regulations are followed when doing his assignments and he also wants to make sure that there is no room for accusations of favoritism. All the drivers like to have the long runs so these must be fairly distributed to avoid jealousy arising among the drivers.

4.2.4 Approach

The approach to the design of the Dispatcher's Assistant was to first implement an effective aid which would reduce the dispatcher's heavy workload. Any assistance which a system could provide would certainly reduce the dispatcher's stress and result in more efficient and safe delivery of the product. If some of the more routine aspects of dispatching could be achieved by the system, the dispatcher would have more time to work in areas where his skill and experience could be better put to use. If the assignment of the drivers can be done efficiently and fairly it should result in increased driver satisfaction, safer delivery of the product and increased customer satisfaction.

There are a great number of administrative details and calculations which have to be performed, and if these are completed with pencil and paper they represent a very significant load on the person. Yet, many of these procedural tasks are intimately connected to the intellectual task and can not be effectively assigned to an assistant. By removing the burden of having to concentrate on these calculations and the strain of having to remember or search through papers or documents for many important details, such as the volume of gas that a particular trailer can carry, the dispatcher should be able to do the assignments in less time and be able to concentrate on more important and influential factors which affect the safe delivery of the product.

The aim therefore was to see what assistance could be provided to the dispatcher to relieve the workload and reduce the stress while still maintaining the customer satisfaction and cost-efficient delivery of the product. Any assistance to the dispatcher would have far-reaching effects on the entire delivery process. Relieving stress on the dispatcher is certain to affect the drivers and this in turn is reflected in their attitude and approach to the job and the safe delivery of the product.

The system demonstrates the type of help that can be made available to the dispatcher. Many variations for a production implementation can be thought of and should be specified in a close interaction between user and designer. The availability of a demonstration system makes such an interaction much more concrete and realistic and validates design targets and specifications.

4.3 System Organization

The structure used in the design of the Dispatcher's Assistant can be outlined as follows:

(A) Basic input data
(B) Creation of an internal database
(C) Operations on basic data
(D) Presentation of data to the user
(E) User inputs
(F) Operations on user inputs
(G) Final data presentations

(A) Basic input data

The basic input data are contained in a number of files which are read by the knowledge base. One file lists the customers and includes factual information such as the product each customer uses, the capacity of the customers tank, and the delivery zone in which each customer is located. Another file lists the drivers and related information such as whether he is a senior driver, his preference for gas to deliver, and if a senior driver, which tractor and trailer have been assigned to him. Another file contains the inventory of tractors and trailers, which tractors are sometimes paired with which trailers, and which trailers carry which gas.

For this prototype the basic input data is considered static and changes would be implemented by editing the appropriate files. In an actual implementation this would still be sufficient for the driver, equipment and customer lists. The required delivery amounts, however, are provided by a daily printout from an existing forecasting system and would ideally be transferred directly to the Dispatcher's Assistant.

(B) Creation of an internal database

An internal data base is created from the basic input data using the Class, Object and Property structures supported by NEXPERT. These NEXPERT structures can be seen as the equivalent of relations (tables), records and fields of a relational data base.

The knowledge base developed by the development team contains the framework for the data. However, the values required for an individual working session are loaded from input files. The knowledge base is therefore data independent; a requirement for a system if it is to be easily adaptable to other applications of a similar nature.

Since the amount of input data required for this prototype system is relatively small, the input files are loaded in total. For a production system handling large volumes of data, a selection of the data required during all or part of a working session can easily be implemented as part of the reading process. If appropriate, basic external data can be maintained in a relational data base and the input selection can be handled through query statements issued by the knowledge base.

(C) Operations on basic data

After the basic input data has been transferred to the internal data structures, it is manipulated by the data related operations provided by the expert system. These include subset selections based on comparisons and filtering several levels deep, in NEXPERT called "pattern-matching". Calculations are performed on one type of property value, which depends on selections based on another type. For example, the tanks of industrial customers are allowed to be three-quarters empty before they are due for a delivery while those of a hospital can only be two-thirds empty. Through the process of "pattern-matching" determined by rules in the knowledge base, the actual amount of product required to be delivered to the various customers can be determined using different calculations for the different lists created.

During the progression of a session, originally empty data fields are assigned values as these become known. A driver is assigned to deliver to particular customers with a particular tractor and trailer. The trailer assigned will be selected from a list of trailers which carry the required liquefied gas, which are operational, and which are still unassigned. The driver will be selected from the list of drivers which are on duty for that particular day and shift and are still available for assignment.

In the course of operation several files are written to and read from disk for internal system use. Specifically, the exchange of data with separate programs written in C is accomplished through disk files.

(D) Presentation of data to the user

During a session the user has frequent optional access to a number of reports listing customers, trailers, drivers, etc. These are presented in the form of lists or tables which can be scrolled where required. Where the user is required to input data, the information necessary to make decisions is presented on the same screen display. Whenever possible the user is provided with the option of entering the data by typing a response or by making a choice from a pop-up window display. These screens have been designed using the facilities of the NEXPERT system which provide a rich environment for the choice of colors, formats and display modes.

(E) User inputs

Emphasis was placed on the design of the user interface including the presentation of data to the user and the methods of obtaining user input. In most instances user input does not require any typing other than the scanning of lists with arrow keys and the initiation of actions through control keys. For this purpose, information is presented in the form of scanned multiple choice lists, sometimes in the form of pop-up menus. For instances where the required information is not in the system, for example, the name of a temporary driver, an editing capability is provided. Tables, in which entries can be edited, are typically used to provide the user with an overview before finalization. This gives an opportunity for a last change, if required. With every screen an appropriate prompt message explaining the required action is displayed.

(F) Operations on user inputs

After a user has completed the input(s) requested by a particular screen display and terminates with the "continue" command (F1 key), the internal data and status of the system is updated and the inferencing process continues until the user is presented with new data or other inputs are required. In one instance it was found that instantaneous feedback to the user was necessary. In this instance, it was the incremental total amount of product which had been selected to be delivered by a driver. This instantaneous feedback was best accomplished by writing a program in C for a customized screen display which performed the required calculations and updated values without leaving the screen.

(G) Final data presentations

Final data is presented in the form of six reports which can be individually viewed and/or optionally printed. The report entitled FINAL REPORT can be printed as customer assignments for individual drivers, replacing the present manual writing. This report, as can all of the reports, can be printed and used by the dispatcher as the assignment sheets for his drivers for the shift. It includes which gas he will deliver, the number of the tractor and trailer he will use for the run and the list of customers to whom he is assigned to deliver product. Although adequate formatting capabilities are provided in NEXPERT, it was decided to supplement these with a C subroutine, utilizing the convenience of calling externals from NEXPERT. Nevertheless, this report too can be previewed from the terminal without terminating the NEXPERT session.

The complete list of reports available are:

(a) FINAL REPORT
(b) SELECTED DRIVERS
(c) GAS TRAILER
(d) ALL TRAILERS
(e) CUSTOMERS
(f) NO REPORTS

A complete description and example of these reports is provided in Section 7. Operations Manual.

5. Structure of the Knowledge Base

5.1 Knowledge Base Files

A collection of rules constitutes a knowledge base (KB). This set of rules can be used by a control mechanism called a Knowledge Processor to intelligently and dynamically browse through the knowledge. The links between the rules reflect the actual structure of the knowledge.

Primarily for work assignment reasons, the knowledge base consists of two files with the arbitrarily chosen names AAB.TKB and CTX327.TKB as their DOS file names. These files are consecutively loaded for execution and contain the rules and data structures required to run a session.

Data read from external files and from the terminal are treated as dynamic data and cease to exist within the knowledge base after the respective outputs have been produced and the dispatching session has been terminated. Data which should be preserved is written back into disk files. The rules, data structures and objects functioning as variables are permanently contained in the knowledge base which is also a set of files executable by the system.

5.2 Data Files

The actual data required by the knowledge base is contained in database files with an NXP extension which are readable by the expert system. This simple file format has been found to be totally satisfactory for this prototype application and would probably also suffice for a product implementation of this project. If required, however, the capability of NEXPERT to access large external databases can be used.

NXP files are also used to store intermediate results and to transfer data to and from external programs.

The data is processed according to the actions specified in the rules during the execution of the program. In some instances it was found to be more efficient to use an external program and NEXPERT provides statements for the execution of such programs. In this project three external C programs have been written to combine and format data and for a special interactive screen display.

The final output of the system is a printout of delivery instructions for each driver for whom assignments have been made. This is one instance where a C program is used to re-format data written by NEXPERT in order to obtain additional flexibility in formatting.

Intermediate data is frequently available during the course of a session for the optional review, editing or printout by the user.

6. Principles of Operation of the Expert System

The operation of the Dispatcher's Assistant from a user's point of view is described in 7. **Operations Manual**. This section gives a more technical explanation of the rules and data structures used.

6.1 Knowledge Base

After loading of the knowledge base and initialization of the system, data on the drivers is retrieved from a database file. This is a file with the "NXP" extension, which is a proprietary database format from Neuron Data with which easy read and write access can be achieved. Information such as classification as to senior or non-senior, preference for type of gas, tractor assigned if senior, and the schedule of drivers assigned to day, afternoon and night shifts for the week are loaded into the system through activation of the initial rule. These data are internally stored as dynamic objects (created at runtime) belonging to the class called "driver_class". The user is given the opportunity to edit the driver schedule for any changes which may have occurred in the schedule (eg. driver is sick) since the schedule was originally created. The day of the week and the shift for which scheduling will be performed must then be selected. The user inputs the date for which he wishes to do scheduling. This may be different from the day's system date.

At this point the user has optional access to screens showing driver assignments, trailers available and information on customers, the product they use and the volume of product their tank contains. User input triggers rules in the knowledge base which in turn causes a display of the appropriate screen files.

Two NXP files are written, one containing driver data, the other the input date which is the date for which the scheduling is to be done. An external program is then executed to combine driver data with the primary list of trailers. Although this action can be performed by rules within NEXPERT, it was found to be more expedient to apply a program written in C. The combined list is read and the data is loaded into the class "trailers".

The user has an opportunity to edit the list of trailers. This editing capability would be used primarily to exclude trailers which for some reason are temporarily unavailable. A sub-class containing only available trailers is created.

The list of customers to be used for assignments is loaded into "customer_class". Each of the customers in the database becomes an object in the knowledge base. Again, it is created dynamically and ceases to exist at the end of the session. Figure 7 shows the object editor in NEXPERT Development version. A customer, AJAXHOSPO, becomes an "object" in NEXPERT when it is read in from the database. The "+" in front of the name indicates it has been created dynamically. It belongs to the NEXPERT class called "customer_class" and has several "properties" such as the type of gas, the amount of product to be delivered, and whether this customer requires delivery. The properties of these objects can be used in the rules for determining what actions will be taken on which objects.

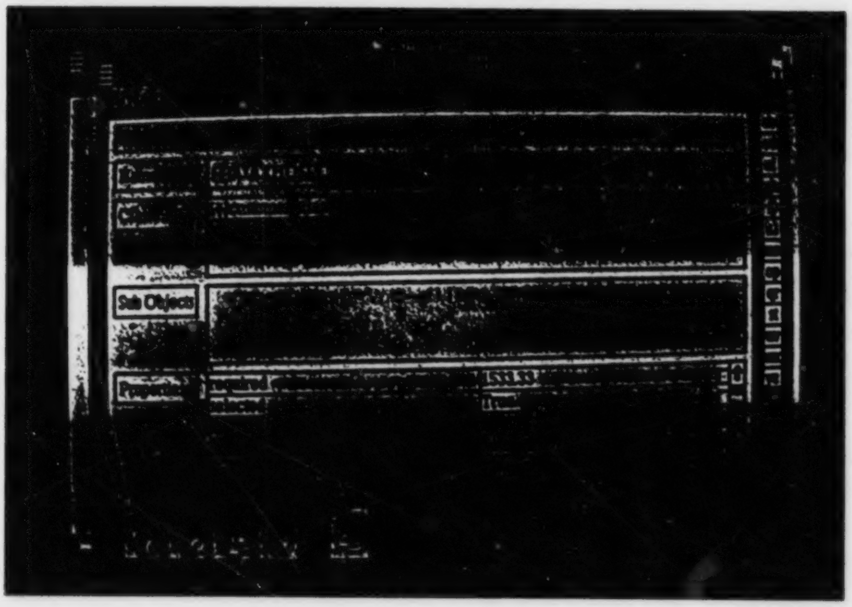


Figure 7 Object Editor

The user is then asked to select the gas which will be concentrated on for scheduling delivery. A list of the four (liquefied) gases handled by the dispatcher is presented and selections are made by using the arrow keys, as in most other instances.

The existing forecasting program provides the dispatcher with a print-out of the customers who require delivery of the product. For the prototype, the customers due for delivery are selected from a multiple choice list displayed on the screen. This selection is made by pressing the arrow keys to move the cursor to the various customers and by pressing the space bar to make the selection.

The expected delivery amounts are calculated by the system using a formula based on the customer's tank capacity and on the type of customer (industrial or hospital). It is assumed that these amounts are correct for the delivery due dates which are provided by an existing forecasting program.

The completion of the above selection, as well as the completion of any other action, will activate appropriate rules which in turn initiates subsequent actions. In this case, a sub-class of "selected_customers" is created which is then further subdivided by geographical zone. This information on geographical zone was originally given in the initial customer data. The total delivery requirement for each zone is then calculated.

Another rule causes the sub-selection of trailers available for delivery in accordance with the gas which has been selected as the gas to concentrate on for delivery.

A screen presentation shows the total amount of product required to be delivered in each zone and requests selection of the first zone to be scheduled. The list of customers which require delivery for the selected zone and the list of available trailers for the selected gas are written to temporary files and this is used to provide input to the assignment module. This module has been written in C in order to provide an immediate response, as opposed to a few seconds of response time if implemented through the knowledge base.

Presented next, on the left of the screen is the list of customers which require delivery in the zone which has been selected for assignment and, on the right is the list of trailers which are available for delivery. These lists, like any others presented by the system, are scrollable if the data exceeds the size of the presentation window.

The user can now select any number of customers from the customer window and one (and only one) trailer from the trailer window, Figure 8. The total amount of product to be delivered in this selected zone is displayed below the customer window. As the user selects customers from the list, the total volume of product selected to be delivered is also displayed and the zone total is reduced by that same amount. Both customer and trailer selections can be changed at will. When one trailer and the customers to receive delivery by this trailer have been selected, <Alt><A> is pressed on the keyboard to confirm the selection and the customers and trailer selected are highlighted on the screen by a color change.

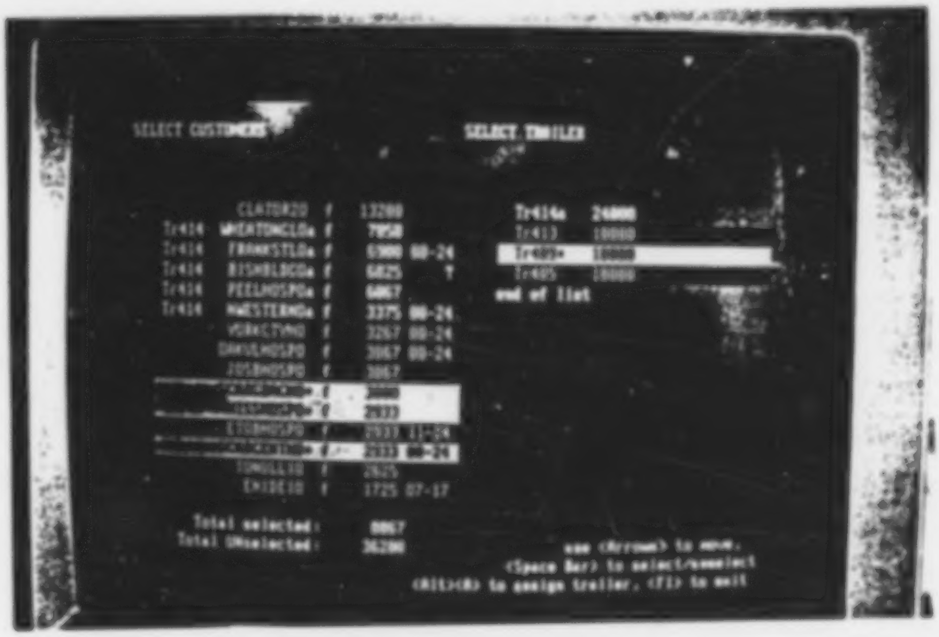
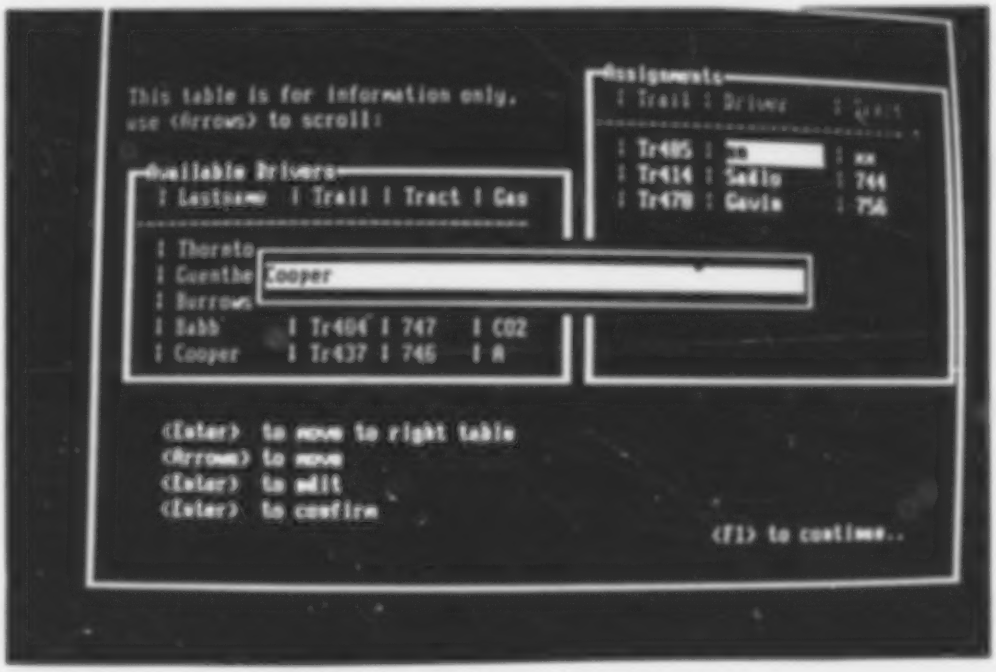


Figure 8 Assigning Customers and Trailers

This process is repeated until all customers in the selected zone have been assigned to trailers. There will however, be an opportunity later to fill in any voids if the assignment is left incomplete. Pressing <F1> causes the zone selection screen to re-appear for selection of the next zone to be assigned and the process can be repeated until customers for each zone have been assigned. The total amount required to be delivered in each of the as yet unselected zones is shown.

After the customer-trailer assignment is completed, drivers and tractors will be assigned to the trailers. Certain tractors are permanently assigned to certain drivers and there is a trailer which is usually assigned with that tractor. The combination of driver and trailer will be shown in the displayed data.

The driver assignment screen again has two windows, one with all available but unassigned drivers and, if applicable, their regular trailers and/or tractors. This window is used for reference and cannot be edited. The other window shows the selected trailers and the drivers for the permanently assigned trailers.



```
This table is for information only.
use <Arrows> to scroll:

Available Drivers                                Assignments
 | Lastname   | Trail  | Tract | Gas              | Trail | Driver    | Tract
 ----------------------------------              -----------------------------
 | Thornto                                        | Tr405 | Co        | xx
 | Guenthe  Cooper                                | Tr414 | Sadlo     | 744
 | Burrows                                        | Tr470 | Gavin     | 756
 | Babb       | Tr404 | 747   | CO2
 | Cooper     | Tr437 | 746   | A

   <Enter>  to move to right table
   <Arrows> to move
   <Enter>  to edit
   <Enter>  to confirm
                                              <F1> to continue..
```

Figure 9 Driver Assignment

Any entry in this trailer - driver - tractor table can be edited and this feature is used to fill in unassigned drivers. See Figure 9. Some tractors can be selected from the reference list. To fill in the remaining tractors the user will make his or her selection from a printed list - in this prototype implementation.

Finally, a summary list of customers, their assigned trailer, driver and tractor, is brought to the screen for a final review and changes. Any customers not yet assigned for delivery can be dealt with here. The <F1> key again represents "ok" and data is written to a temporary file. From this temporary file the formatting for the printed driver assignment sheets is performed by a C program. The user again has an opportunity to review on screen, but not to change this time, a number of reports which are generated. The available reports include the driver assignments. A printout of any report can be initiated at this time.

Having completed the dispatch assignment for one type of gas, the user can now repeat the process for the other gases to be delivered.

During the course of a session the following screens are presented.

A. Information only type of displays:

1. "DISPATCHING ASSISTANT"

B. Interactive screens requiring user input

2. "DRIVER SCHEDULE"
3. "SCHEDULING"
4. "IF YOU WANT TO SEE REPORTS"
5. "CHANGES IN TRAILER AVAILABILITY"
6. "CHANGE AVAILABILITY OF TRAILERS"
7. "SELECT THE TYPE OF GAS"
8. "SELECT CUSTOMERS FOR DELIVERY"
9. "REQUIRED DELIVERY VOLUMES PER ZONE"
10. "TRAILER/CUSTOMER ASSIGNMENT"
11. "ASSIGNED TRAILERS/AVAIL DRIVERS"

6.2 Computing Resources and Software

Expert systems are heavy users of computing resources. Previously known reputable expert systems shells used powerful workstations as their platform. The developers of NEXPERT Object have managed to design a very capable system which can be executed on many platforms from mainframes to Macintosh and IBM compatible PCs.

This project has been implemented on a 386 PC IBM compatible with initially a total of 2MB of main memory. On the PC platform, memory utilization is still a major concern since the DOS operating system allows direct access only to 640K. In order to utilize the 1.4 MB of extended memory, a special "memory manager" program has to be applied and QEMM-386 by Quarterdeck was used for this purpose.

Overall, a significant amount of work was spent on systems configuration and optimization and on the installation of several versions of the software as upgrades became available. Eventually the expected performance and response time as well as a very robust operation was achieved for the development version of NEXPERT, a factor crucial for the project development since excessively slow response times and frequent crashes are counter productive.

A different set of system problems had to be resolved for the delivery system under the run time version of NEXPERT. Here, initially the execution of programs external to NEXPERT but called from the knowledge base could not be achieved. The cause was analyzed as insufficient memory available below 640K and the recommended solution, to acquire the "protected mode" version of the runtime system, was followed.

"Protected Mode" is a mode of operation available on the 286 and 386 type microprocessors which allows direct addressing of memory beyond 1MB, as opposed to the normal or "Real Mode" which is limited to 1MB by the microprocessors and to 640K by DOS. With protected mode, most of the NEXPERT code resides above 1MB leaving the lower memory free for other programmes which are concurrently resident.

Use of the Protected Mode version of NEXPERT Runtime not only made the external programs operational but also caused a remarkable improvement in the speed of execution and response time.

Unfortunately, the availability of the Protected Mode version came late in the project and forced earlier work to be diverted to the experimentation with potential alternate solutions. This however, is not uncommon for work at the leading edge of technology, where both the tools and methods are still being refined.

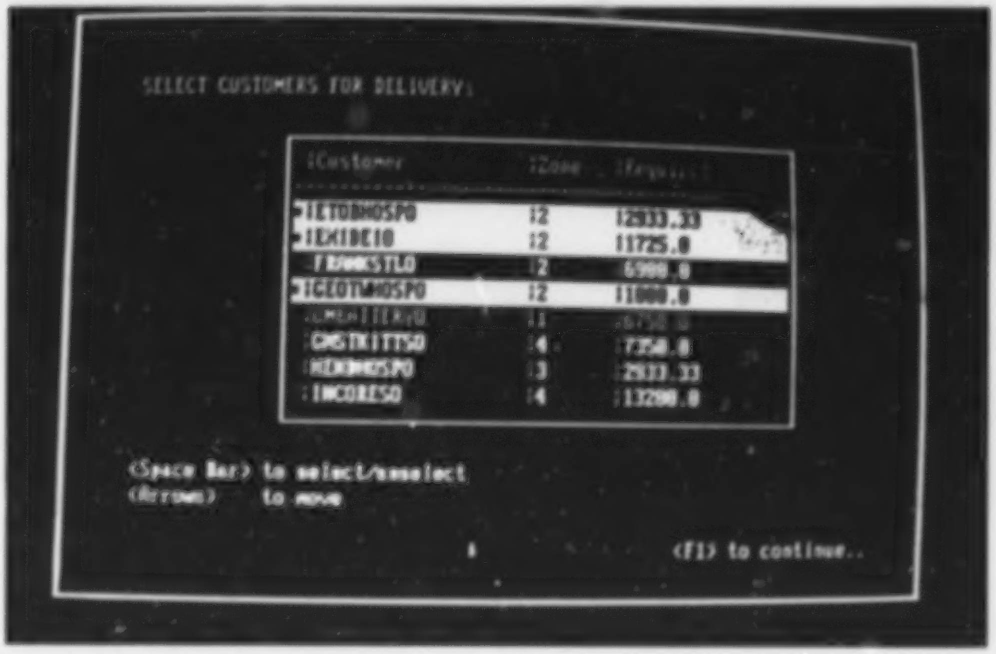
6.3 Visual and Keyboard Interface

The visual user interface is designed as "screens" which appear on the terminal. These screens can be designed to provide information to the user or to retrieve some information necessary. Information required to be input can be entered either through a menu or by typing or saying a response to a query. These screens, or "forms" as they are called in NEXPERT, are contained in two types of files. These files may have an "FRM" or "TXT" extension and they are compiled at run time.

The "FRM" screens may or may not be interactive with the user depending on their function and design. Certain screens allow for editing of data, some allow only a single choice to be made, some allow multiple choices to be made from a list and some allow new data to be entered directly. The "TXT" screens are for giving information to the user and they cannot be edited or altered by the user during runtime.

In designing the screens, special attention was paid to the design of a "friendly" interface between the system and the person operating it. All requests for input are presented on color screen displays with functional layouts and a pleasant choice of color. Information presented on a screen is relevant to the task at the time and the screens are designed so that the information required for that task is available and the user does not have to recall what had been entered on a previous screen. But, this information needs to be presented in such a way that it is not overwhelming and confusing. A graphical or tabular means of presentation is easier to read and extracts information and explains a solution clearer than a strictly textual presentation.

Where appropriate, the screen interface has been designed with multiple choice selections where the user can select as many or as few choices as required. See Figure 10. In filling in forms or answering questions, there are occasions when typing the answer is the best alternative and there are times when a multiple choice or pop-up menu is the best choice, Figure 11. It is important to recognize where each of these should be used. For example, if a typing mistake will require many entries to be re-done, then it is preferable to have a choice menu where the chance of error is greatly reduced.

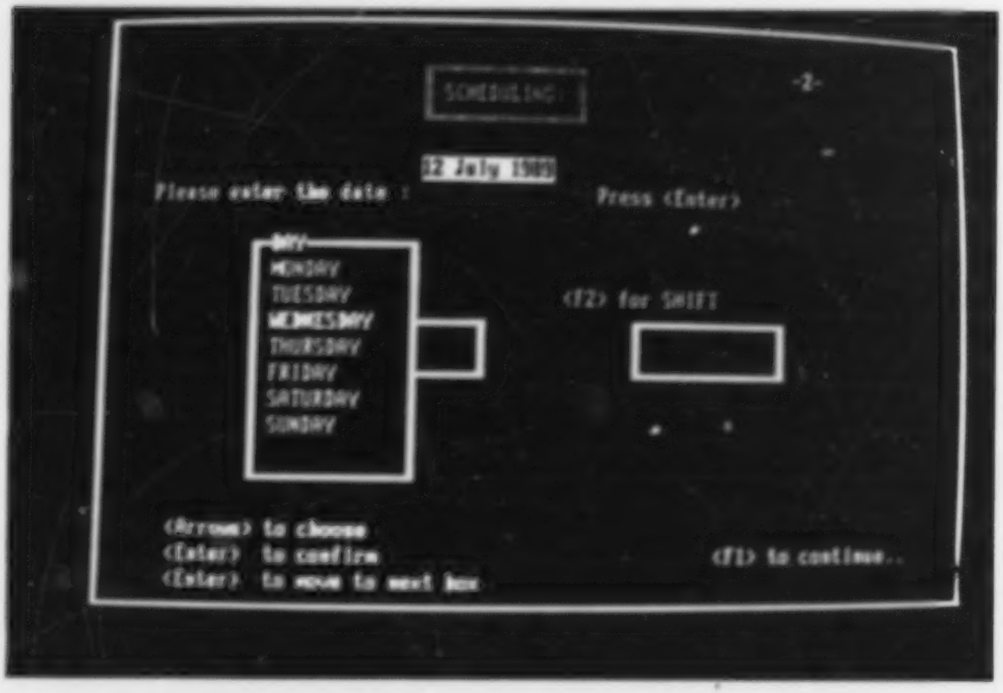


```
SELECT CUSTOMERS FOR DELIVERY:

        :Customer          :Zone  :Required
        ---------------------------------------
       >:ETOBHOSP0         :2     :2933.33
       >:EXIDE10           :2     :11725.0
        :FRANKSTL0         :2     :6900.0
       >:GEOTWHOSP0        :2     :11000.0
        :[illegible]       :1     :6750.0
        :GMSTKITTS0        :4     :7350.0
        :HEXXHOSP0         :3     :2933.33
        :INCORES0          :4     :13200.0

<Space Bar> to select/unselect
<Arrows>    to move

                                        <F1> to continue..
```

Figure 10 Example Multiple Choice Screen

A major design consideration was consistency in the use of key functions and the avoidance of illogical actions such as "press Enter key twice". The one external program providing an important user interaction was designed to adhere to the conventions given by the built-in facilities of NEXPERT.

Screen displays of reports are available at several steps in the process. Where such reports exceed the area of one display screen they are scrollable through the use of the arrow keys.



Please enter the date : 12 July 1989

Press <Enter>

<F2> for SHIFT

<Arrows> to choose
<Enter> to confirm
<Enter> to move to next box

<F1> to continue..

Figure 11 Example Pop-Up Screen

6.4 Voice Interface

In addition to the normal keyboard-screen operation of the application a hands-free interface based on a speech recognition system and a speech synthesizer has been implemented.

For speech input the DragonKEY module of the VoiceScribe 1000 system by Dragon Systems, Inc. is used. For this speaker dependent recognition system the vocabulary necessary to operate the Dispatchers Assistant prototype was defined. To the extent possible the screen prompts of the keyboard system were used: e.g. where a text prompt appearing on the screen reads "To continue press F1" the spoken word "Continue" was programmed to be recognized and to activate the function of the F1 key. "Up", "Down", "Left", "Right" replaces the arrow keys, "Enter" the Enter key, etc.

The defined vocabulary must be "trained" for the operators voice. This is accomplished using the features of the VoiceScribe system. Several users can be trained and command files written to allow the switch between users and the training of new users to be initiated with keyboard commands like "Train Lisa" or "Speaker George".

Synthesized speech output is produced by the ACCENT system of Aicom Corp. It establishes a DOS device on the PC to which text strings can be written and are translated into voice output. Again, the prompt text appearing on the screen is typically spoken by the synthesizer. Since the NEXPERT Runtime version does not provide a facility to write directly to a DOS device, a small external program is called with a different command line parameter for every voice message.

Both the recognizer and synthesizer system use their own plug-in board in the PC in addition to their software systems with connections for a microphone and speaker, respectively.

7. Operations Manual

The Dispatcher's Assistant developed by Applied AI Systems, Inc., uses knowledge-based system technology in order to greatly speed up the entire process of assigning drivers and equipment to particular gas shipments. A technical description of the system's operation can be found elsewhere. This manual is intended to give the user a practical step-by-step guide to operating the Dispatcher's Assistant.

The config.sys file may have to be adjusted for memory and file configuration. Please see Appendix B for the proper configuration file and the loading of batch files.

7.1 Installing the System

1. Create a new directory on the computer's hard drive into which you will load the system.

   ```
   C:\>MD TDC   (press Enter)
   ```

   Note: The user can select any name, other than TDC, for the directory.

2. Put the system diskette into the A: drive, and copy all its files into the directory you created in step 1.

   ```
   C:\>copy a:*.* C:\TDC
   ```

7.2 Keyboard and Voice Commands

A key innovative feature of the Dispatcher's Assistant is its speech interface system. The movement from one screen to another, navigating within a screen, and entering information as required can be all done by voice commands. This feature gives the dispatcher considerable flexibility when using the system. For each voice command there is a corresponding keyboard entry, and the dispatcher can move easily between voice and keyboard input. Both keyboard and voice commands are included in this section, with voice commands written in quotes and in italics and **keyboard in bold.**

In what follows, we will assume that the system has already undergone the necessary voice recognition training. The system will not recognize new words on its own accord. The system must be trained to recognize each word in its vocabulary of voice commands.

7.3 Starting Up A Session With Dispatcher's Assistant

1. If voice commands are to be used, the system must first be told who the user will be. The name used here must correspond with the name used when the person was trained on the voice system. This command can only be entered at the keyboard.

```
C:\TDC>Speaker Tom
```

Once the voice system has been activated, the keyboard and voice can be used interchangeably. At the beginning of each session, the user must activate the system's Protected Mode Runtime by typing SETRT (which must be done from within the TDC directory).

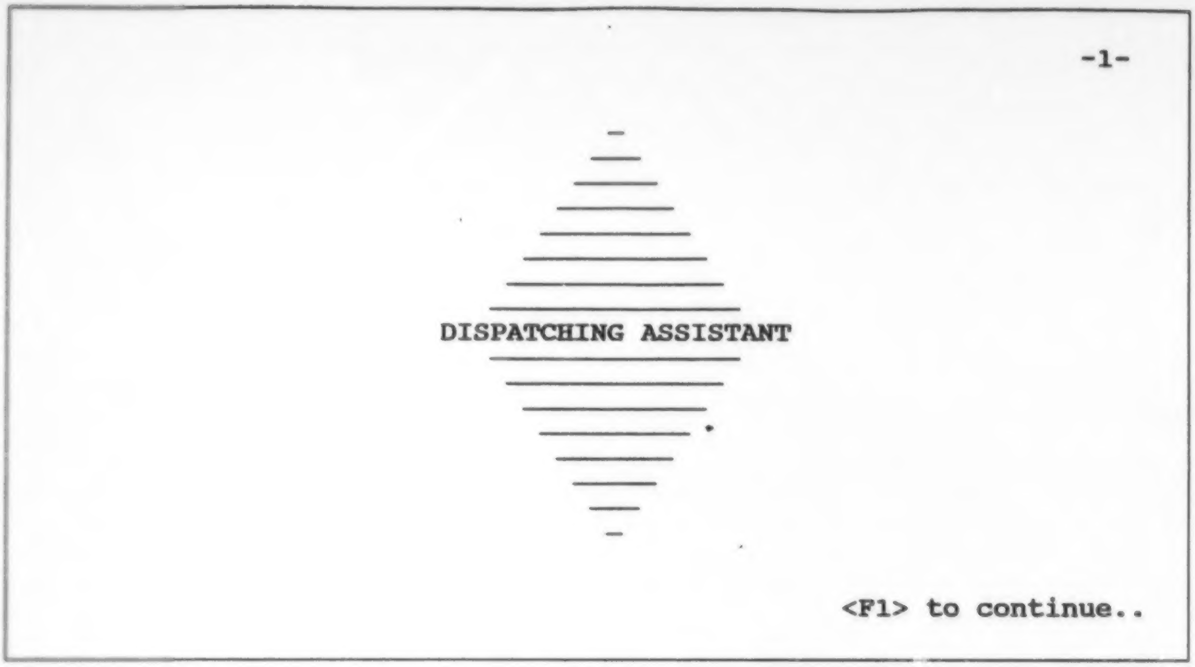
```
C:\TDC>SETRT     (press Enter)

C:\TDC>"SET RT"
```

Note: This step assumes that the user has already installed the NEXPERT Runtime key on the computer's printer port.

2. In order, to initiate the Dispatcher Assistant, type

```
C:\TDC>NX       (press Enter)

C:\TDC>"NX"
```

After the **NX** command, an opening screen, Figure 12, will appear. Press the function key F1 to continue or give the voice command "Continue".


-1-

DISPATCHING ASSISTANT

<F1> to continue..

Figure 12 Opening Screen

The Dispatcher's Assistant organizes the dispatching process through a series of screens. On each screen there is a simple set of instructions indicating the nature of the information to enter, along with instructions on how to go on to the next step. Below is an explanation of each screen. The screens are listed in the order in which they will appear to the user. For a given session, the user may not see all the screens described below. Certain screens will only appear as a consequence of options chosen in previous screens.

7.4 Moving Within a Screen, and From Screen to Screen

The movement of the cursor, within any menu, is controlled by the following voice commands:

"GO UP"

"GO DOWN"

"GO LEFT"

"GO RIGHT"

These commands move the cursor either one line up, or down, or one space to the right, or to the left. In order to cause larger cursor movements, the above commands can, in this version, be augmented by the digits 2 or 4. For example one can issue the voice command "UP 4", or "DOWN 2", etc.

These commands can also be carried out with the **<Arrow>** keys on the keyboard. There is no one keystroke for moving by 2 or 4 units.

To move on to the next screen just say "Continue", or press the function key <F1>.

If for any reason, you want to end in the middle of a session and return to DOS, just say "Control Q" or type **CTRL-Q**.

7.5 DRIVER SCHEDULE Screen

The first screen to appear after the opening TITLE screen, is the DRIVER SCHEDULE screen, Figure 13. This screen contains a list of all the drivers, along with their scheduled shifts for the week. A driver can be in one of four categories:

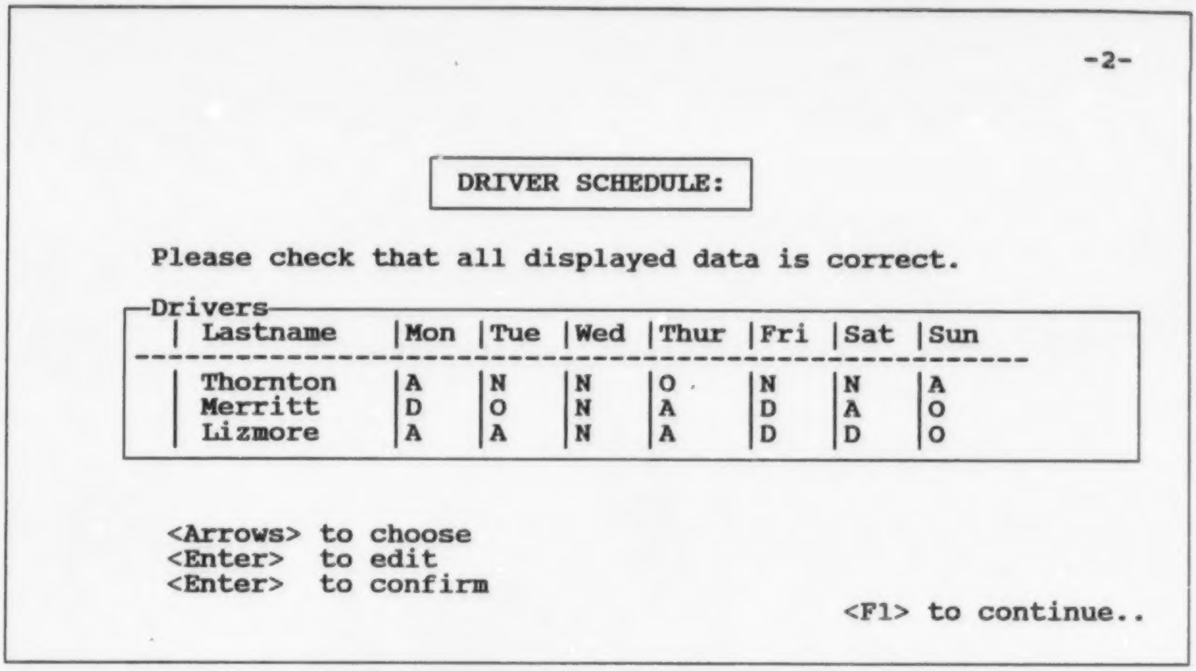
a) **"D"** - working the day shift

b) **"A"** - working the afternoon shift

c) **"N"** - working the night shift

d) **"O"** - off for that day

The purpose of this screen is to ask the user to verify that the names and work schedules are indeed correct. If however, the information is in error or needs to be revised, then changes can be made.



-2-

DRIVER SCHEDULE:

Please check that all displayed data is correct.

Drivers

Lastname	Mon	Tue	Wed	Thur	Fri	Sat	Sun
Thornton	A	N	N	O	N	N	A
Merritt	D	O	N	A	D	A	O
Lizmore	A	A	N	A	D	D	O

<Arrows> to choose
<Enter> to edit
<Enter> to confirm

<F1> to continue..

Figure 13 DRIVER SCHEDULE Screen

7.6 Editing the DRIVER SCHEDULE SCREEN

In order to edit any information, position the cursor over the piece of information that has to be changed. To start the editing:

<Enter>

"Enter".

A pop-up editing box will appear where the editing is done. To make a change in the shift say "Afternoon", "Day", "Night", or "Off", depending on the new shift which is to be assigned to the driver. In order to confirm this change, the voice command "Enter" puts the change into the table.

The same changes can be made via the keyboard using the **<Arrow>** keys to select the item. To edit the item, press **<Enter>** and then type in the correct information, using **A**, **D**, **N**, **O** for the afternoon, day, and night shift and day off, respectively. Press **<Enter>** again in order to record the change.

In the present prototype, only some of the drivers names have been entered. A complete list of the words trained in the vocabulary is included in Appendix C. Changes in names can be entered from the keyboard by typing in the name.

While keyboard entries can easily produce typing errors voice commands are more reliable. Thus, when using the key board, be careful in entering any changes. Certain error checking facilities have not been included in this prototype. As result, typing in any letters other than **D**, **A**, **N**, and **O**, will not be detected. Such a misprint will carry through the system and cause some confusion.

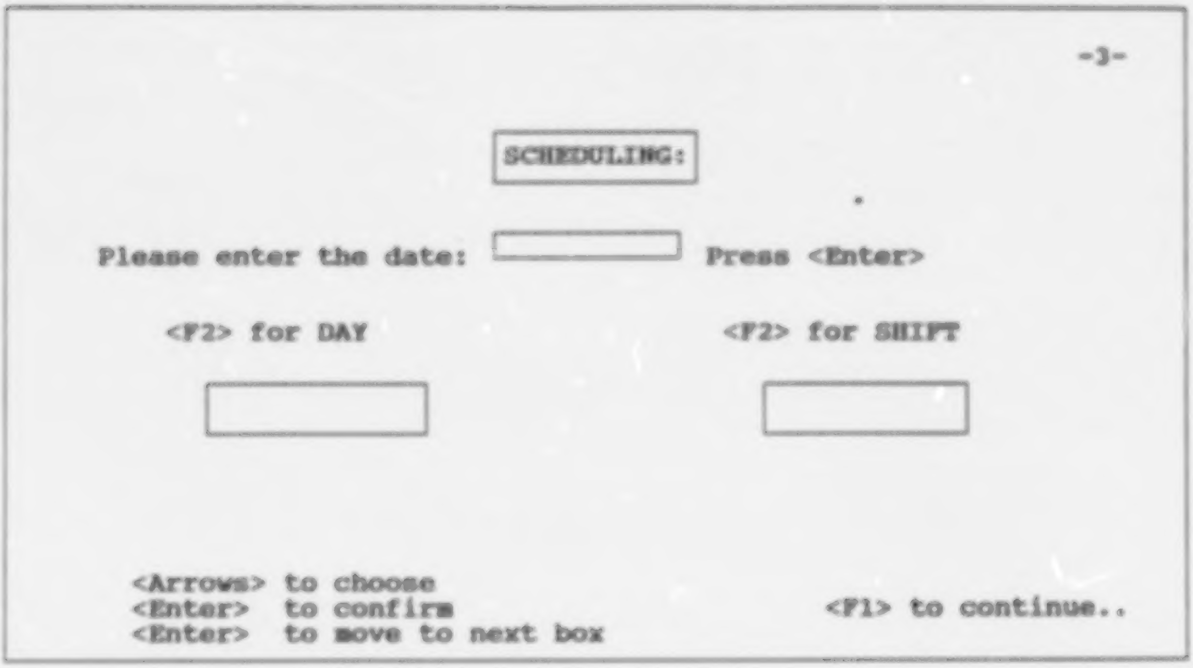
7.7 SCHEDULING Screen

In this screen, the dispatcher selects the particular date, day, and shift for which delivery assignments will be made.

The entire dispatching process is carried out separately for each shift in the day, and for one day at a time. The cursor is first located at the "Please enter the date" category in the screen. See Figure 14. To enter the date for which you want to make the dispatching assignments, say the particular date and then say "Enter". The date must adhere to the format Month, Day, Year. Upon saying "Enter" the cursor moves to the DAY category. Whereupon, the command "Show List" brings up a pop-up menu showing the days of the week. Move the cursor to the appropriate day and then say "Enter". A similar procedure occurs for the SHIFT category.

Once again, when using the keyboard, the format and correct spelling are important. The month must be written out, either in full or abbreviated form.

By pressing the function key **<F2>**, in either the DAY or SHIFT categories, the user is presented with selection from which to choose. To select any item, use the **<Arrow>** keys, and then press **<Enter>**.



-3-

SCHEDULING:

Please enter the date: Press <Enter>

<F2> for DAY

<F2> for SHIFT

<Arrows> to choose
<Enter> to confirm
<Enter> to move to next box

<F1> to continue..

Figure 14 Day/Shift Scheduling

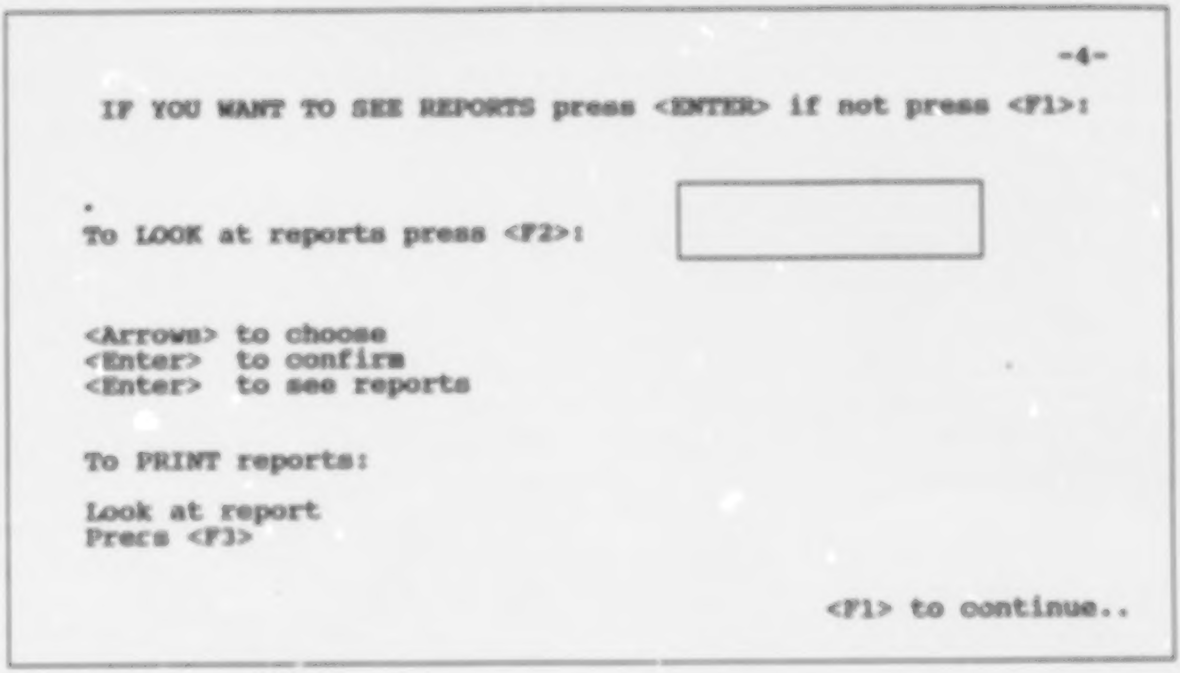
7.8 See Reports

The user has the opportunity to see any of the available reports at this time. See Figure 15. If you do not want to see reports simply say "Continue" or press <F1>. If you wish to see reports, say

"Enter"
"Show list".

Select the report you wish to see by positioning the cursor and say

"Enter"
"Continue".



```
                                                        -4-
  IF YOU WANT TO SEE REPORTS press <ENTER> if not press <F1>:

 .
 To LOOK at reports press <F2>:


 <Arrows> to choose
 <Enter>  to confirm
 <Enter>  to see reports

 To PRINT reports:

 Look at report
 Prers <F3>

                                            <F1> to continue..
```

Figure 15 Interim Reports

To view reports using the keyboard method, press

<Enter>
<F2>.

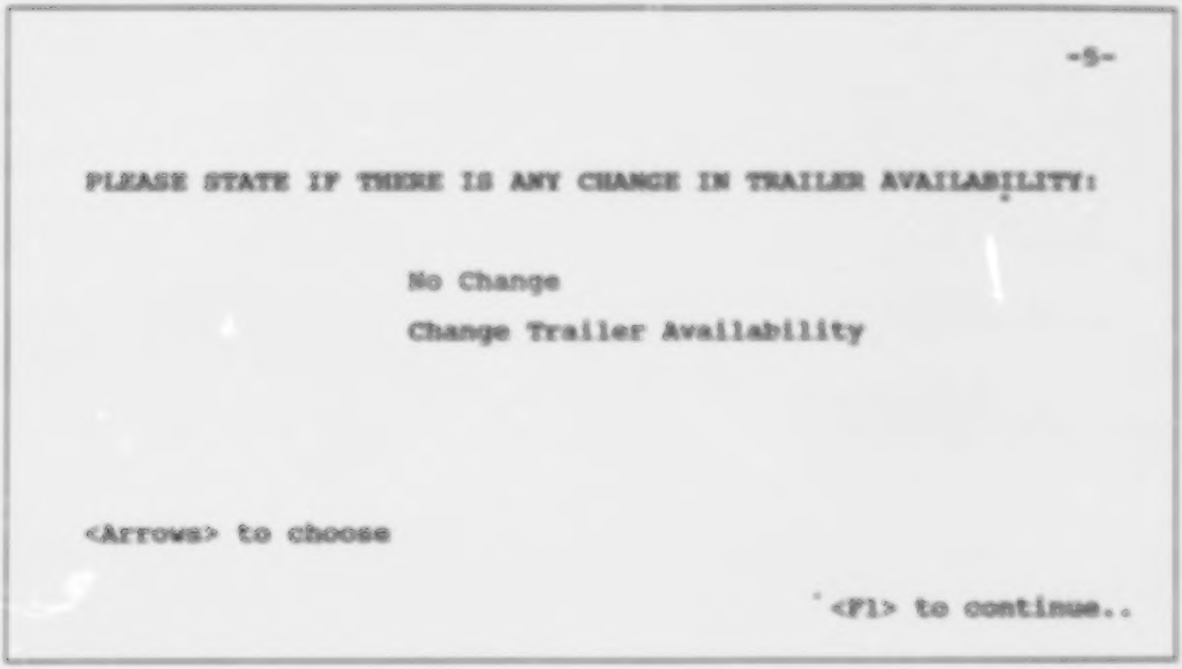
Select the report which you want to see by positioning the cursor with the arrow key, press

<Enter>
<F1>

After viewing the report "Continue" or <F1> will bring you back to the original screen. You have the opportunity to look at any other report.

7.9 TRAILER AVAILABILITY Screen

In this screen, the dispatcher is given the opportunity to make changes in any information concerning the availability of particular trailers. The cursor is placed on the appropriate selection, Figure 16, and either press <F1> or say "Continue" to proceed to the next screen.



-5-

PLEASE STATE IF THERE IS ANY CHANGE IN TRAILER AVAILABILITY:

No Change

Change Trailer Availability

<Arrows> to choose

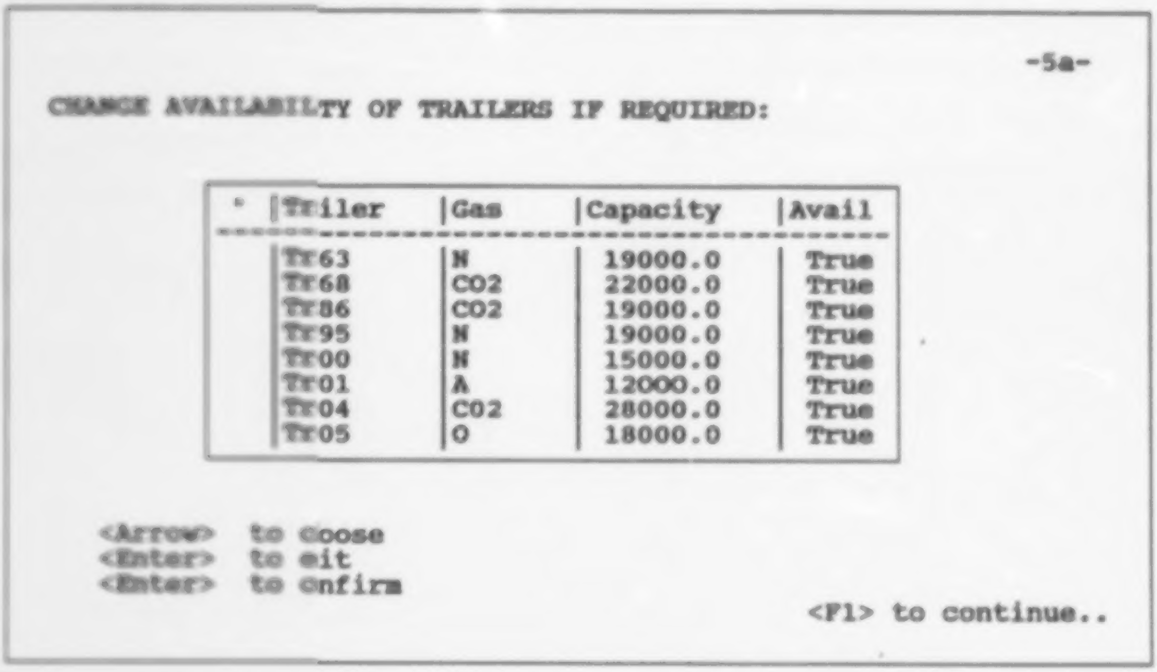
<F1> to continue..

Figure 16 Change Trailer Availability

If you select "No Change", the system jumps to screen 6 (TYPE OF GAS). If you select "Change Trailer Availability", then the "CHANGE AVAILABILITY" screen appears.

7.10 CHANGE AVAILABILITY Screen

In this screen eah trailer in the fleet is listed along with the type of as it carries and its capacity. There is also a cdumn indicating whether the trailer is available or not. See Figure 17. On the screen, this is indicated by eithr True or False.



```
                                                      -5a-
CHANGE AVAILABILTY OF TRAILERS IF REQUIRED:

          * |Trailer   |Gas     |Capacity   |Avail
          ---------------------------------------
            |Tr63      |N       |  19000.0  | True
            |Tr68      |CO2     |  22000.0  | True
            |Tr86      |CO2     |  19000.0  | True
            |Tr95      |N       |  19000.0  | True
            |Tr00      |N       |  15000.0  | True
            |Tr01      |A       |  12000.0  | True
            |Tr04      |CO2     |  28000.0  | True
            |Tr05      |O       |  18000.0  | True

   <Arrow>  to coose
   <Enter>  to eit
   <Enter>  to onfirm
                                        <F1> to continue..
```

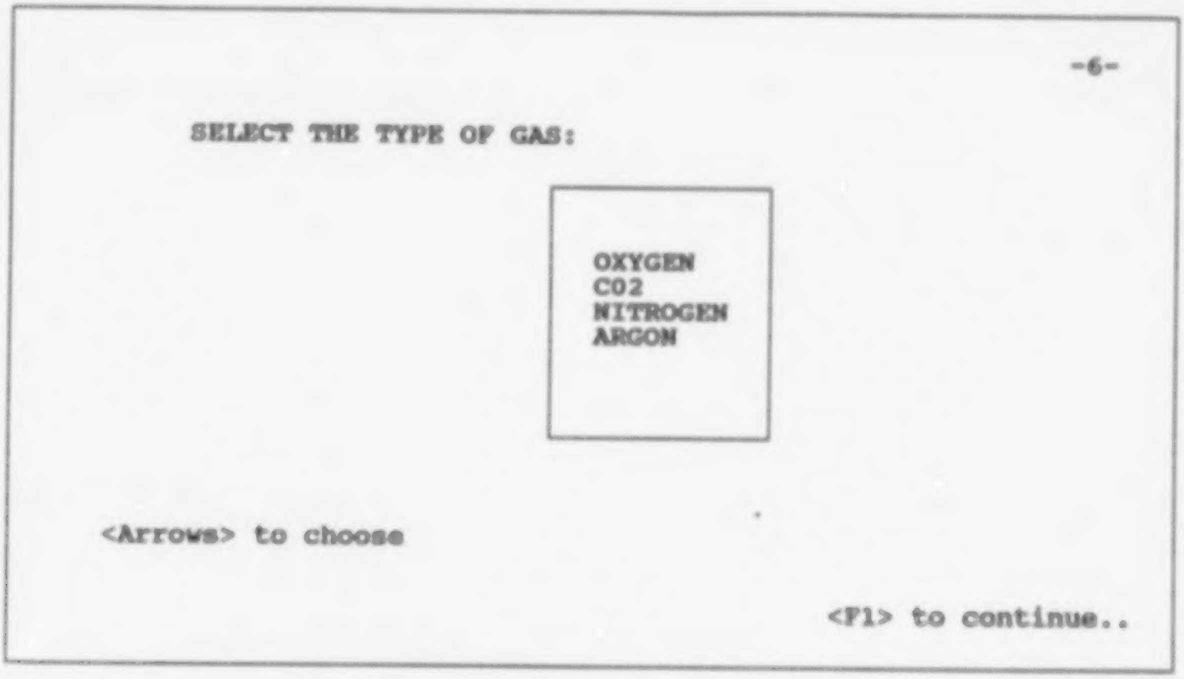
Figure 7 List of Available Trailers

To change the avalability of any given trailer, move the cursor to the rowindicating the trailer in question and then move the curor to the "Avail" column. One could just as easily goto the column first and then down to the row. Onze the curor is in the right position, say either "Available" or "Nt Available".

Making changes through the keyboard follows the same procedure as in the other screens, except than one types in True to signify available, and False for the opposite.

7.11 **TYPE OF GAS Screen**

In this screen the dispatcher selects the type of gas which he currently wishes to focus on for scheduling his drivers to deliver. See Figure 18.



-6-

SELECT THE TYPE OF GAS:

OXYGEN
CO2
NITROGEN
ARGON

<Arrows> to choose

<F1> to continue..

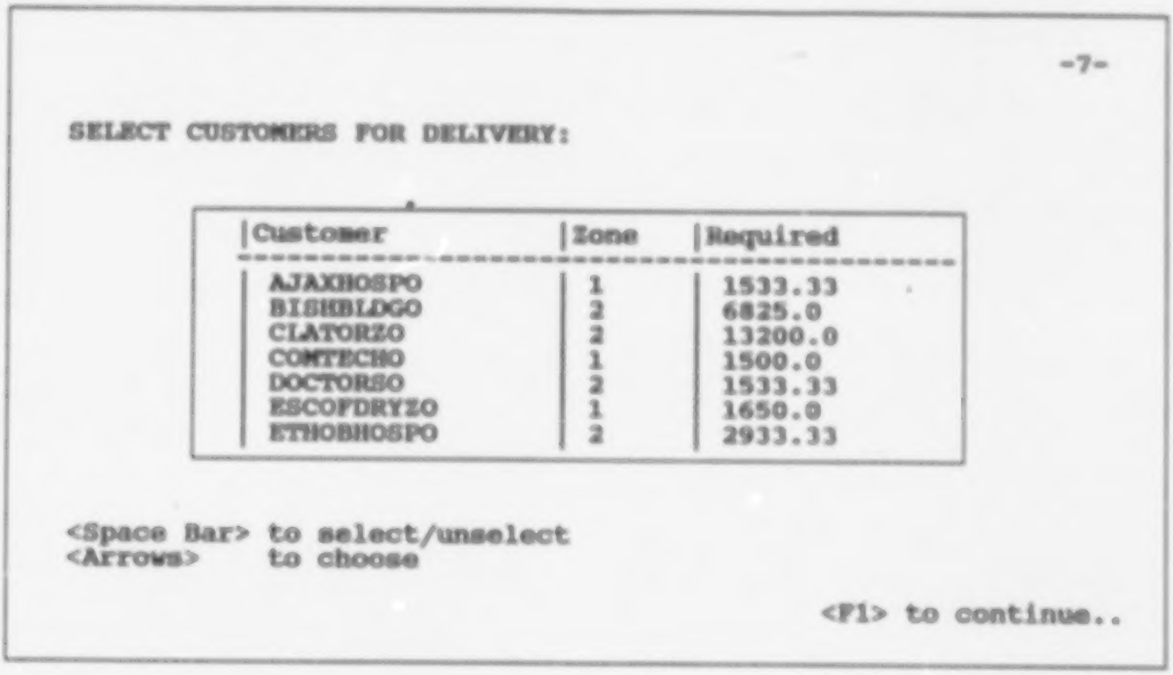
Figure 18 Type of Gas

Each gas is scheduled as a separate session, i.e. the entire dispatching process is carried out entirely for one gas before moving on to the next.

The gas can be selected by moving the cursor to the correct gas and selecting it either by saying "Continue" or pressing <F1> at the keyboard.

7.12 SELECT CUSTOMERS Screen

In this screen, Figure 19, all the company's customers who use the type of gas that was selected on the previous screen are listed, along with the zone in which they are located and the volume of gas they require.



-7-

SELECT CUSTOMERS FOR DELIVERY:

Customer	Zone	Required
AJAXHOSPO	1	1533.33
BISHBLDGO	2	6825.0
CLATORZO	2	13200.0
COMTECHO	1	1500.0
DOCTORSO	2	1533.33
ESCOFDRYZO	1	1650.0
ETHOBHOSPO	2	2933.33

<Space Bar> to select/unselect
<Arrows>   to choose

<F1> to continue..

Figure 19 Select Customers

The volume of gas require is specified in the "Required" column. The type of gas is indicated by a one letter suffix added to the customer's name, e.g. COMTECHO, means COMTECH requires liquid oxygen.

From this screen, the dispatcher will select all the customers to which deliveries will be made for that day. His selection of customers is based on a print-out from a computerized forecasting system which is already installed.

To select the customer move the cursor to the name and say

"Select".

With the keyboard, selection is made by pressing the **spacebar**.

If a mistake is made and a selection should be canceled, just move the cursor to the name and say

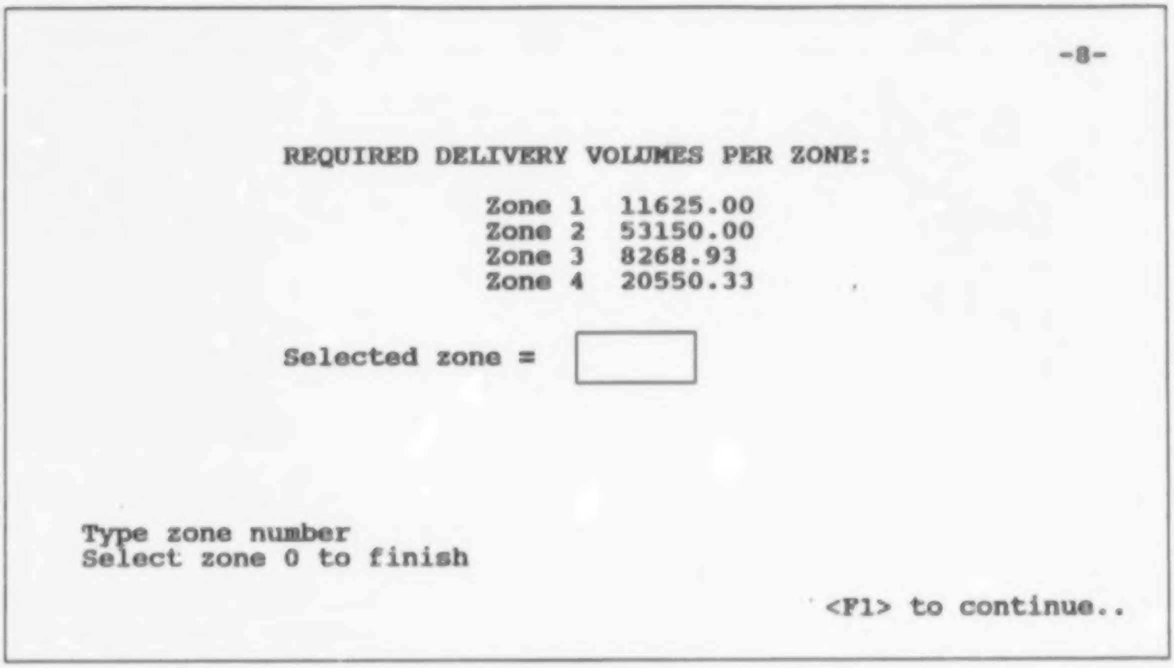
"Unselect".

Or, press the **spacebar** again to unselect.

Once all the required customers have been selected, say "Continue" , or type <F1>, to move on to the next screen.

7.13 **REQUIRED DELIVERY VOLUMES PER ZONE Screen**

Based on the selections in the previous screen, the next screen, Figure 20, shows the calculated amount of the particular liquid gas that is required to be delivered in each zone.



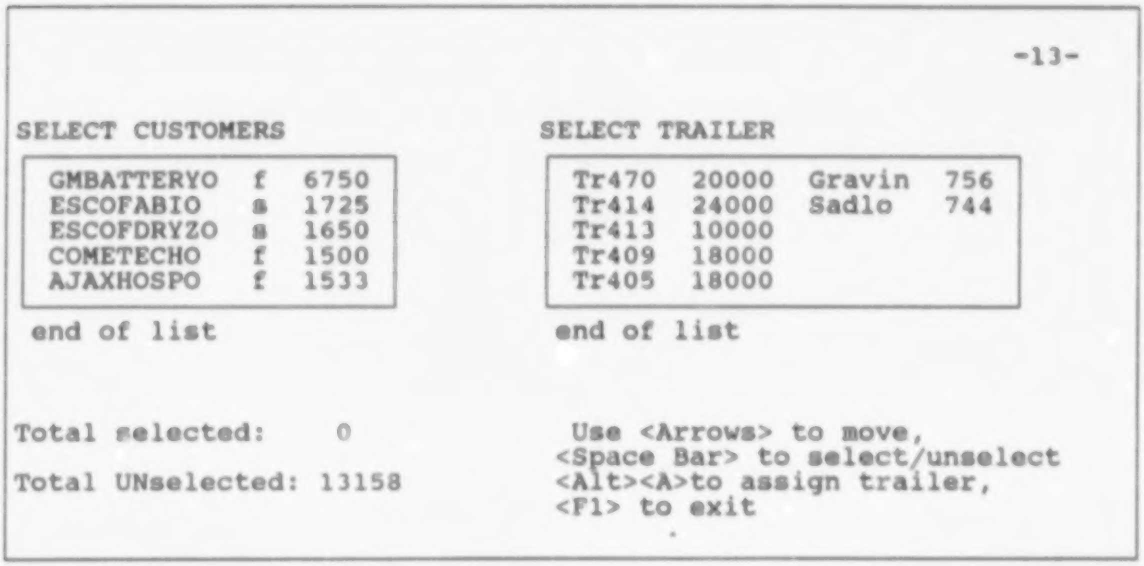
```
                                                           -8-

              REQUIRED DELIVERY VOLUMES PER ZONE:

                        Zone 1  11625.00
                        Zone 2  53150.00
                        Zone 3  8268.93
                        Zone 4  20550.33

              Selected zone =



  Type zone number
  Select zone 0 to finish
                                            <F1> to continue..
```

Figure 20 Delivery Volume Per Zone

The zone to which drivers and equipment will be assigned is selected on this screen. To choose a zone merely say "1", or "2" etc., or type in the number. You can start with any zone. Once the selection has been made, say "Continue" or press <F1>.

7.14 SELECT CUSTOMER / SELECT DRIVER Screen

Once a zone is selected, all the customers that have been selected as requiring a delivery of product in that zone are shown on the screen, Figure 21. Also shown are the available tractors and trailers for that particular gas and their drivers, if one of the units has been assigned to a senior driver.



```
                                                              -13-

SELECT CUSTOMERS                          SELECT TRAILER

  GMBATTERYO   f   6750                     Tr470   20000   Gravin   756
  ESCOFABIO    s   1725                     Tr414   24000   Sadlo    744
  ESCOFDRYZO   s   1650                     Tr413   10000
  COMETECHO    f   1500                     Tr409   18000
  AJAXHOSPO    f   1533                     Tr405   18000

 end of list                               end of list


Total selected:     0                       Use <Arrows> to move,
                                           <Space Bar> to select/unselect
Total UNselected: 13158                    <Alt><A>to assign trailer,
                                           <F1> to exit
```

Figure 21 Customers and Equipment

The screen consists of three displays. The left hand window is a list of all the customers for the selected gas, who require delivery of product and fall within the zone selected. The first column in the window is the customer's name, the second column specifies the customer's method of placing orders (c for call in orders, s for systematic orders, and f for orders forecasted by the computer).

The second window (to the right of the first) contains a listing of the trailers available, and their carrying capacities. If there is a senior driver permanently assigned to this trailer, his name is listed in the second column.

At the bottom of the screen are two pieces of information: the total volume of gas for customers already selected, and the volume still left unassigned.

In this screen, the dispatcher assigns a particular delivery trailer to a group of customers. This is accomplished by first selecting a set of customers. Move the cursor to the name and say "Select" or press **<Spacebar>** for selection of the customer. A customer is indicated as being selected by an "*" which appears beside their name. As customers are selected, a running total of their gas requirements is presented in the bottom window. The selection of customers is done with a careful eye to matching total delivery volume to the size of the various available trailers.

After the customers are selected go to the second window and choose one of the available trailers. The same voice command apply here as in the first window. "Go Right" is the voice command to go to the second window, or the **right arrow key**.

Once the trailer is selected, the voice command

"Assign"

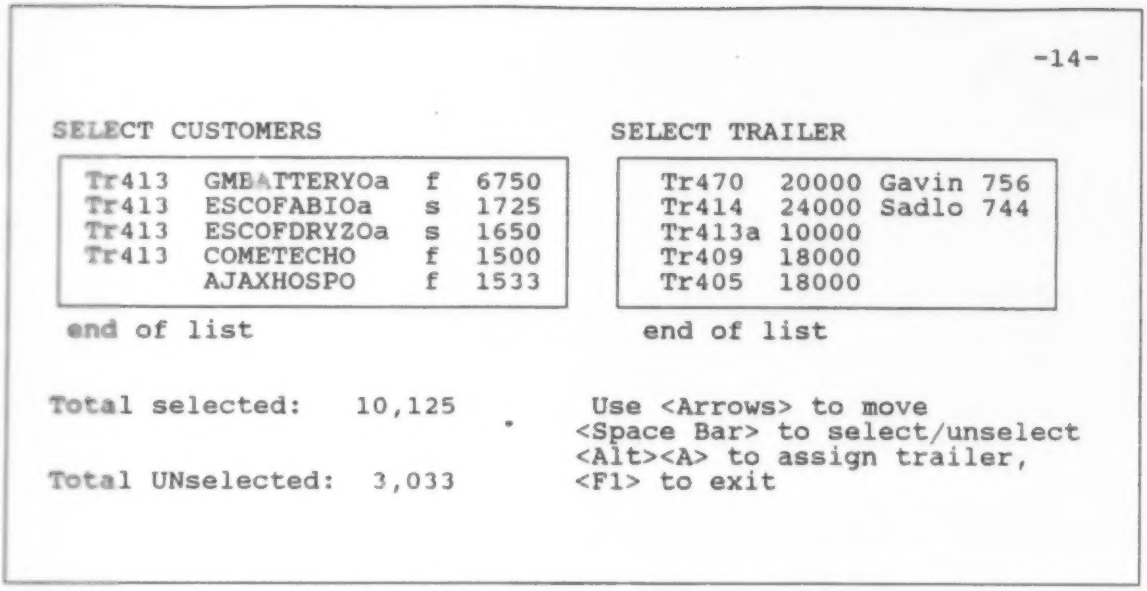
confirms the customers/trailer match-ups, or the keys

<Alt><A>

has the same effect.

After the assignment has been made, an "a" appears after each customers name who has been assigned to a trailer and the color of the customers and the trailer assigned changes to red. See Figure 22.

Repeat the process of assigning customers to trailers until all the liquid gas for that zone has been assigned, or until you do not wish to assign any more customers.



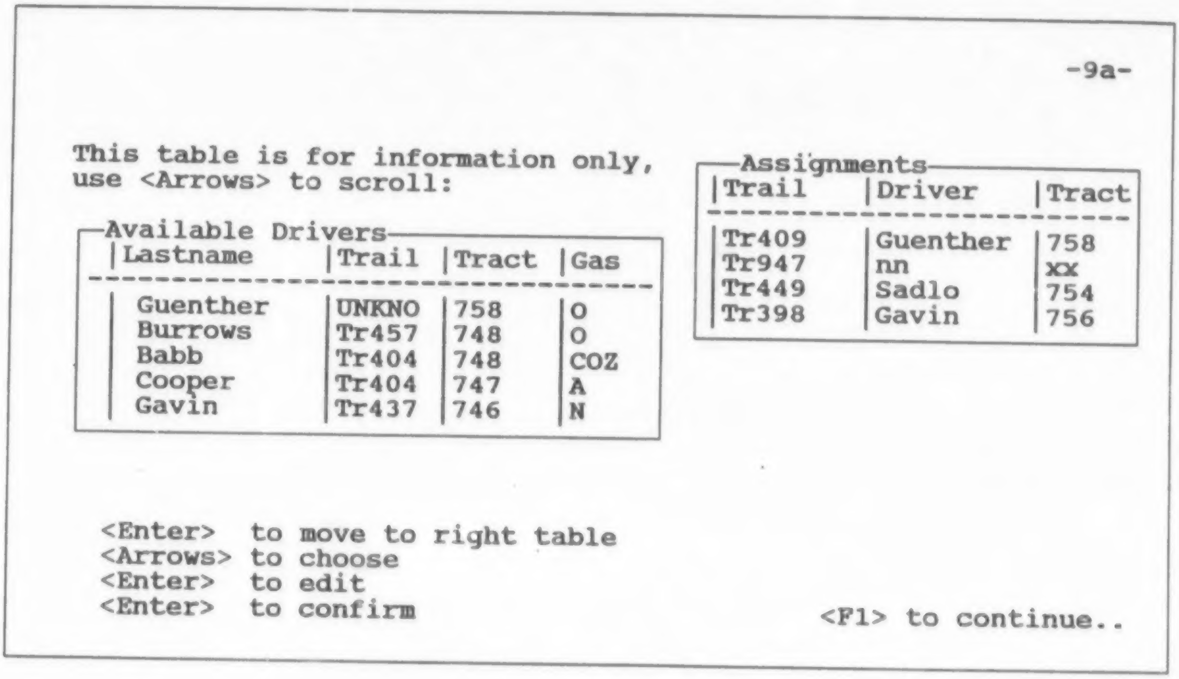
```
                                                          -14-

SELECT CUSTOMERS                            SELECT TRAILER
  Tr413   GMBATTERYOa   f   6750              Tr470   20000 Gavin 756
  Tr413   ESCOFABIOa    s   1725              Tr414   24000 Sadlo 744
  Tr413   ESCOFDRYZOa   s   1650              Tr413a 10000
  Tr413   COMETECHO     f   1500              Tr409   18000
          AJAXHOSPO     f   1533              Tr405   18000
 end of list                                 end of list

Total selected:    10,125           Use <Arrows> to move
                                   <Space Bar> to select/unselect
                                   <Alt><A> to assign trailer,
Total UNselected:  3,033           <F1> to exit
```

Figure 22 Assigned Customers

After all the customers in a zone have been assigned to trailers, each time the system returns to the "REQUIRED VOLUME PER ZONE" screen to choose another zone. Once the process of assigning customers is complete, select "0" and "Continue" or press <F1>.

7.15 Assign Trailers

This screen contains two windows. The left hand window is for information purposes only and it displays which drivers have been assigned to which trailers and tractors, and the gas they are delivering. If a driver has not yet been assigned to a tractor or trailer, UNKNOWN appears in the window beside his name. The senior drivers are automatically assigned to their specific tractor and trailer. See Figure 23. The remaining drivers who were not assigned to trailers will be indicated in this first window. In the second window on the right, one assigns specific drivers and tractors to trailers.



```
                                                                          -9a-

This table is for information only,         ─Assignments─────────────────
use <Arrows> to scroll:                     |Trail     |Driver     |Tract
                                            -------------------------
─Available Drivers──────────────────        |Tr409     |Guenther   |758
  |Lastname      |Trail |Tract |Gas         |Tr947     |nn         |xx
 ---------------------------------          |Tr449     |Sadlo      |754
  |  Guenther    |UNKNO |758   |O           |Tr398     |Gavin      |756
  |  Burrows     |Tr457 |748   |O
  |  Babb        |Tr404 |748   |COZ
  |  Cooper      |Tr404 |747   |A
  |  Gavin       |Tr437 |746   |N


  <Enter>   to move to right table
  <Arrows>  to choose
  <Enter>   to edit
  <Enter>   to confirm                                    <F1> to continue..
```

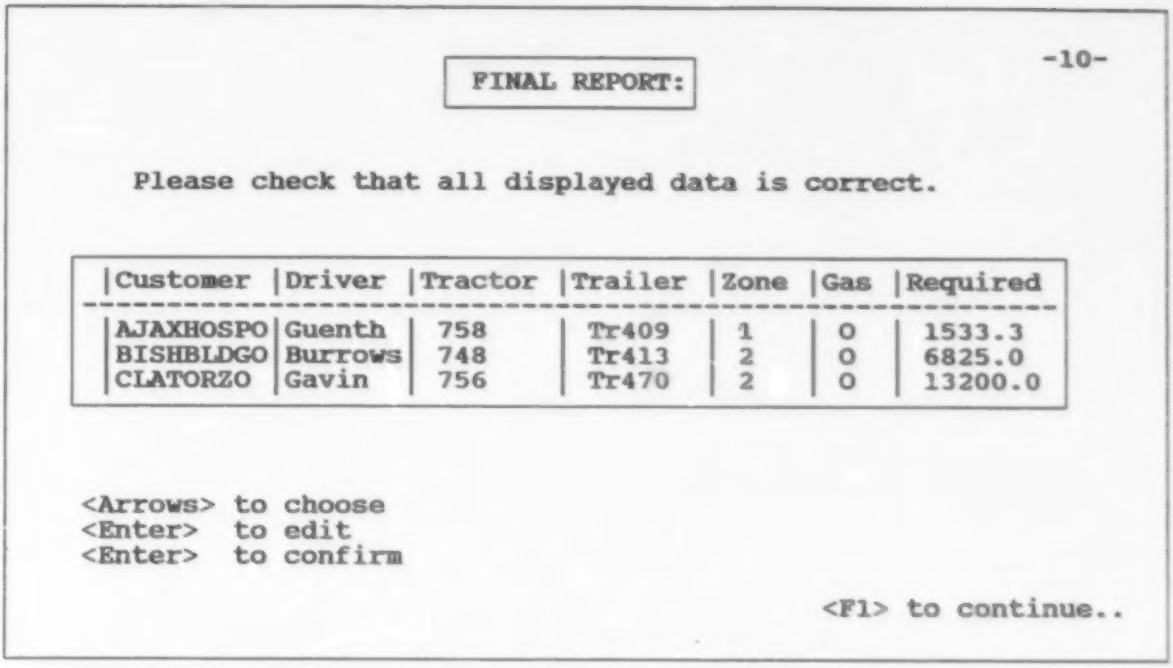
Figure 23 Assigning Drivers to Trailers

To assign drivers and tractors move the cursor to the second window by using the right arrow key or saying "Go Right". Once the cursor is positioned on the row corresponding to the trailer in question, move the cursor to the "Driver" column and say "Enter". At this point an editing window will appear in which the driver's name can be typed in. Say "Enter" again to confirm this choice. Exactly the same procedure is used to assign tractors.

Once all the desired editing has been completed, press <F1> or say "Continue" to go on to the next screen.

7.16 FINAL REPORT Screen

This screen lists the amount and type of gas that will be delivered to each customer, along with the driver, tractor, and trailer that will be doing the delivery.



```
                                                        -10-
                      FINAL REPORT:

     Please check that all displayed data is correct.

  |Customer |Driver |Tractor |Trailer |Zone |Gas |Required
 -----------------------------------------------------------
  |AJAXHOSPO|Guenth | 758    | Tr409  | 1   | O  | 1533.3
  |BISHBLDGO|Burrows| 748    | Tr413  | 2   | O  | 6825.0
  |CLATORZO |Gavin  | 756    | Tr470  | 2   | O  | 13200.0

  <Arrows> to choose
  <Enter>  to edit
  <Enter>  to confirm
                                          <F1> to continue..
```

Figure 24 Assignment Summary

The user is asked to verify the information on the screen. If there are any unexpected events that require last minute changes (a driver getting sick, or tractor breaking down, etc.), then the dispatcher can make the necessary changes.

To edit any particular item move the cursor to the item and say "Enter", then type the correction, and say "Enter" to confirm the change. The key stroke commands are the <**Arrow**> keys to move the cursor, <**Enter**> to edit, and <**Enter**> to confirm.

When satisfied with the report say "Continue", or type <**F1**> to move on to the next screen.

7.17 **SEE REPORTS Screen**

This screen provides the option to view any of the reports which have been created.

If you want to see any of the reports, say,

"Enter" and
"Show List"

to bring up the menu showing the reports available.

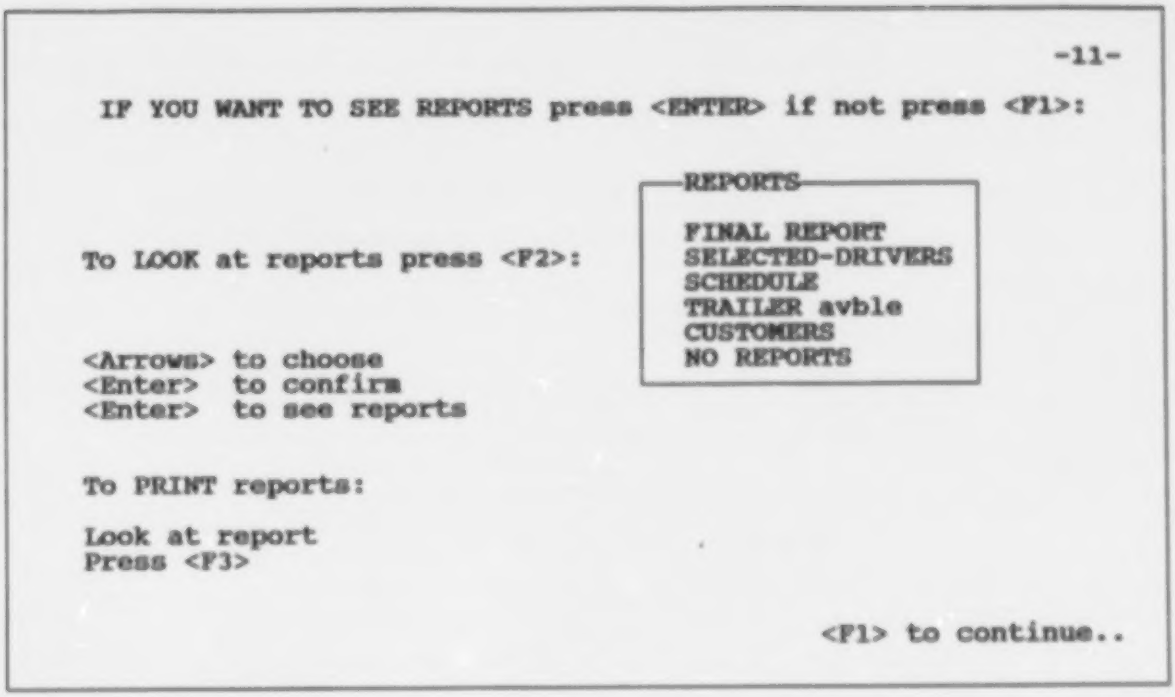
If using the keyboard, press

<**Enter**> and
<**F2**>

The selection of reports, see Figure 25, in the pop-up menu includes:

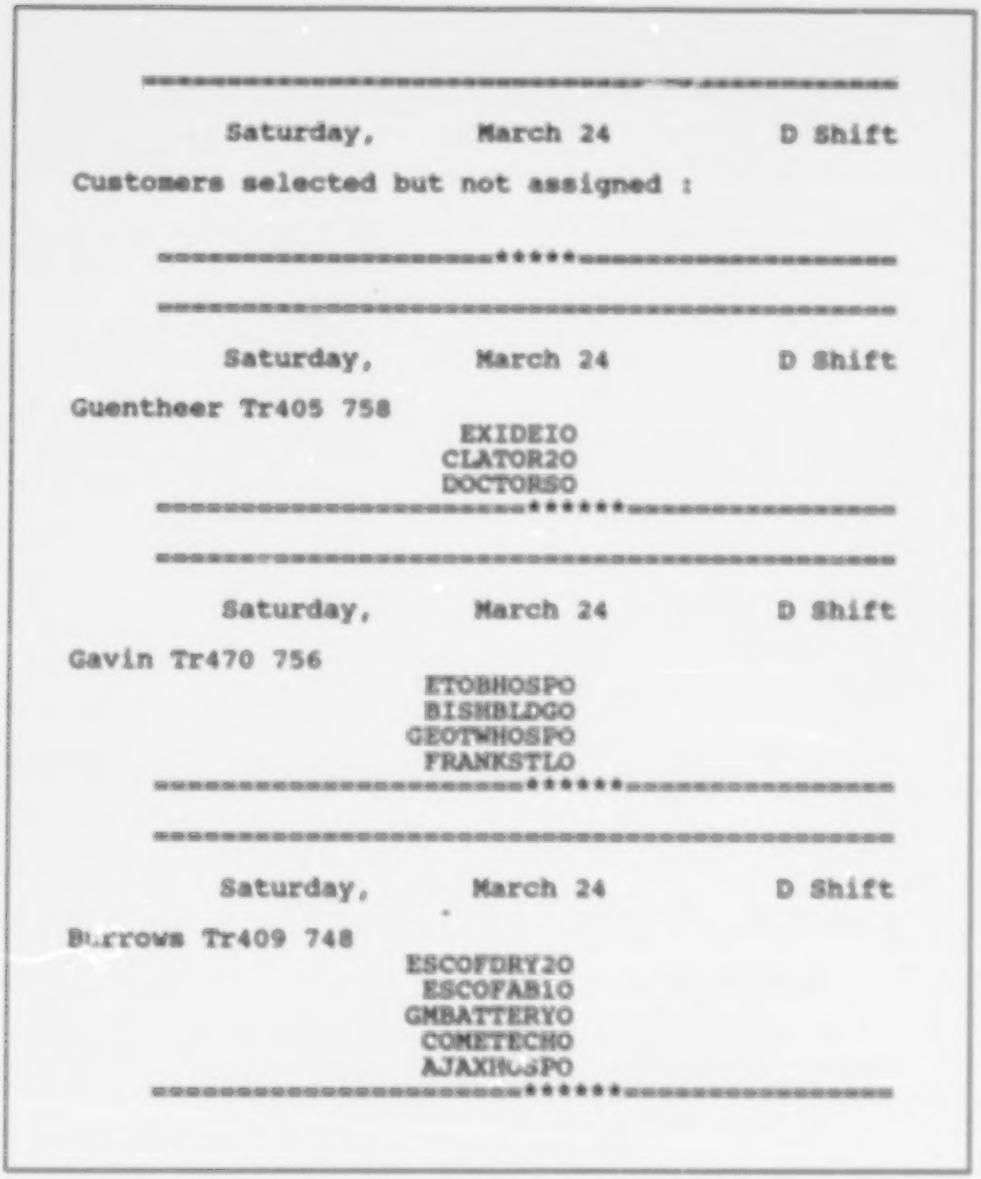
FINAL REPORT (Figure 26)
SELECTED DRIVERS (Figure 27)
TRAILERS FOR PARTICULAR GAS (Figure 28)
ALL TRAILERS (Figure 29)
CUSTOMERS (Figure 30)
NO REPORTS

To view any of the above reports, move the cursor to the desired report and say "Enter" and "Continue" or press <**Enter**> and <**F1**>. When the report is shown on the screen it can be printed by pressing <**F3**>.



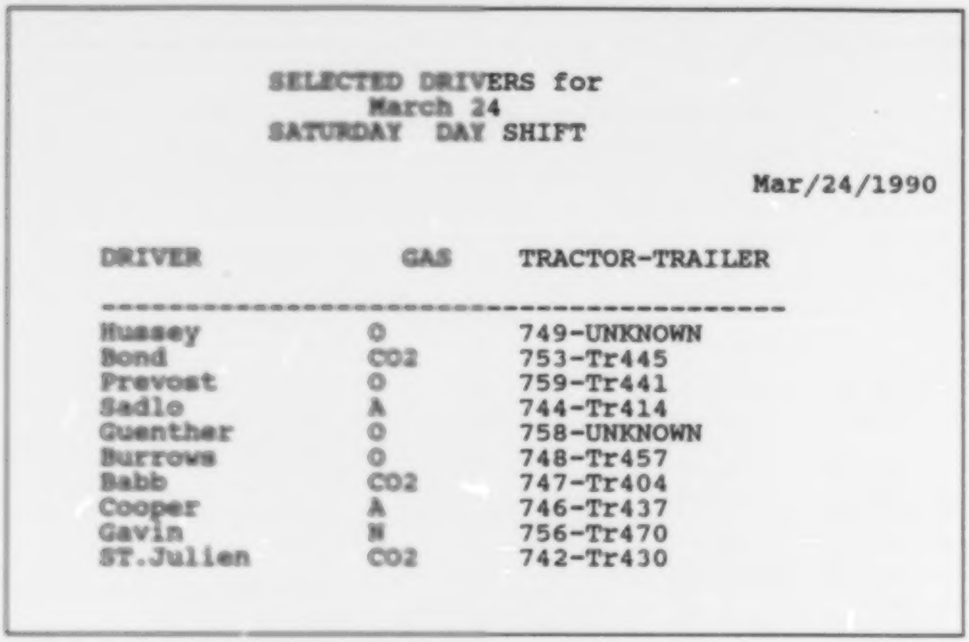
```
                                                                      -11-
     IF YOU WANT TO SEE REPORTS press <ENTER> if not press <F1>:

                                              ┌REPORTS─────────────┐
                                              │                    │
                                              │  FINAL REPORT      │
    To LOOK at reports press <F2>:            │  SELECTED-DRIVERS  │
                                              │  SCHEDULE          │
                                              │  TRAILER avble     │
                                              │  CUSTOMERS         │
    <Arrows> to choose                        │  NO REPORTS        │
    <Enter>  to confirm                       └────────────────────┘
    <Enter>  to see reports


    To PRINT reports:

    Look at report
    Press <F3>

                                                         <F1> to continue..
```

Figure 25 List of Reports Available



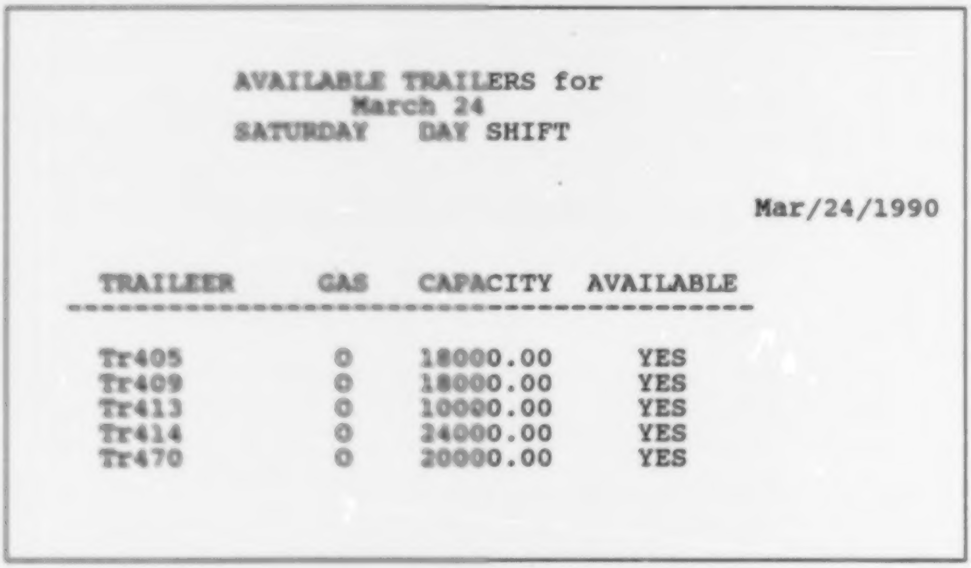
```
     ============================================

         Saturday,      March 24          D Shift

Customers selected but not assigned :

     ====================*****===================

     ============================================

         Saturday,      March 24          D Shift

Guentheer Tr405 758
                        EXIDEIO
                       CLATOR20
                       DOCTORS0
     ======================******================

     ============================================

         Saturday,      March 24          D Shift

Gavin Tr470 756
                      ETOBHOSP0
                      BISHBLDG0
                     GEOTWHOSP0
                      FRANKSTL0
     ======================******================

     ============================================

         Saturday,      March 24          D Shift

Burrows Tr409 748
                     ESCOFDRY20
                      ESCOFAB10
                     GMBATTERY0
                      COMETECH0
                      AJAXHOSP0
     ======================******================
```

Figure 26 FINAL REPORT



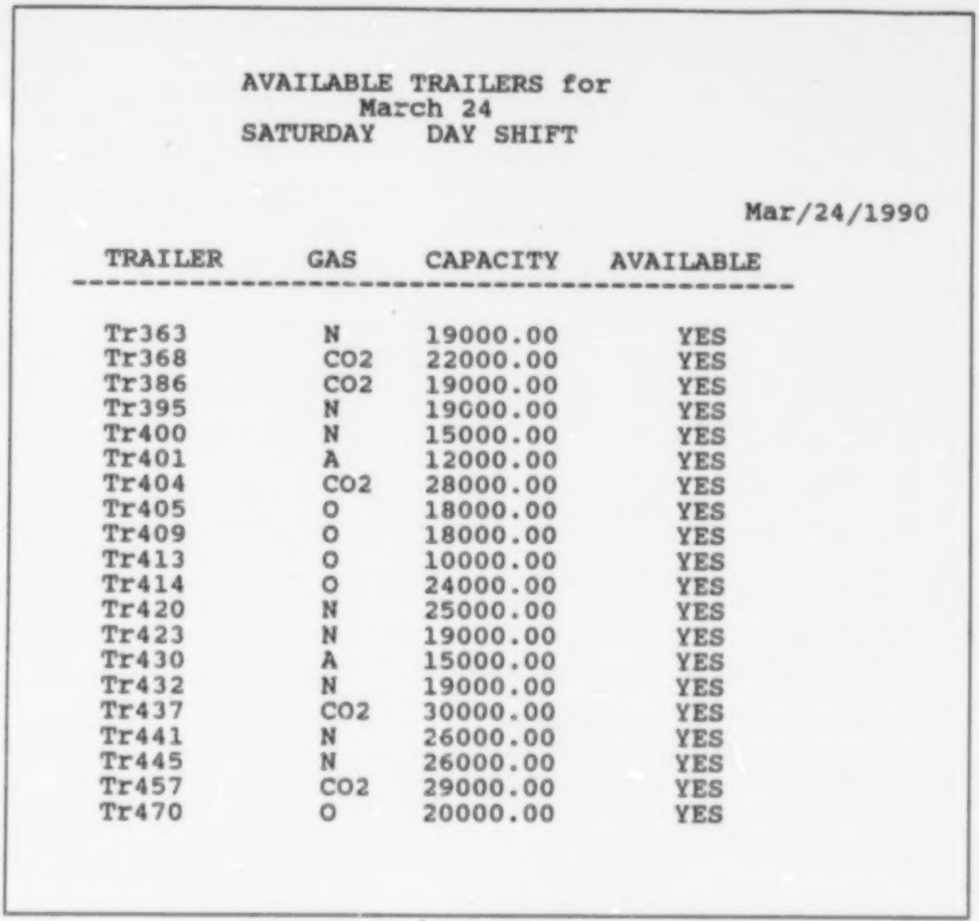
```
              SELECTED DRIVERS for
                   March 24
              SATURDAY  DAY SHIFT

                                          Mar/24/1990

     DRIVER             GAS    TRACTOR-TRAILER

     ----------------------------------------
     Hussey           O        749-UNKNOWN
     Bond             CO2      753-Tr445
     Prevost          O        759-Tr441
     Sadlo            A        744-Tr414
     Guenther         O        758-UNKNOWN
     Burrows          O        748-Tr457
     Babb             CO2      747-Tr404
     Cooper           A        746-Tr437
     Gavin            N        756-Tr470
     ST.Julien        CO2      742-Tr430
```

Figure 27 SELECTED DRIVERS



```
            AVAILABLE TRAILERS for
                  March 24
            SATURDAY   DAY SHIFT

                                          Mar/24/1990

     TRAILEER     GAS   CAPACITY  AVAILABLE
   ----------------------------------------

     Tr405         O    18000.00     YES
     Tr409         O    18000.00     YES
     Tr413         O    10000.00     YES
     Tr414         O    24000.00     YES
     Tr470         O    20000.00     YES
```

Figure 28 Trailers for Particular Gas



```
               AVAILABLE TRAILERS for
                      March 24
               SATURDAY    DAY SHIFT


                                                Mar/24/1990

  TRAILER      GAS    CAPACITY   AVAILABLE
-------------------------------------------

   Tr363       N      19000.00      YES
   Tr368       CO2    22000.00      YES
   Tr386       CO2    19000.00      YES
   Tr395       N      19000.00      YES
   Tr400       N      15000.00      YES
   Tr401       A      12000.00      YES
   Tr404       CO2    28000.00      YES
   Tr405       O      18000.00      YES
   Tr409       O      18000.00      YES
   Tr413       O      10000.00      YES
   Tr414       O      24000.00      YES
   Tr420       N      25000.00      YES
   Tr423       N      19000.00      YES
   Tr430       A      15000.00      YES
   Tr432       N      19000.00      YES
   Tr437       CO2    30000.00      YES
   Tr441       N      26000.00      YES
   Tr445       N      26000.00      YES
   Tr457       CO2    29000.00      YES
   Tr470       O      20000.00      YES
```

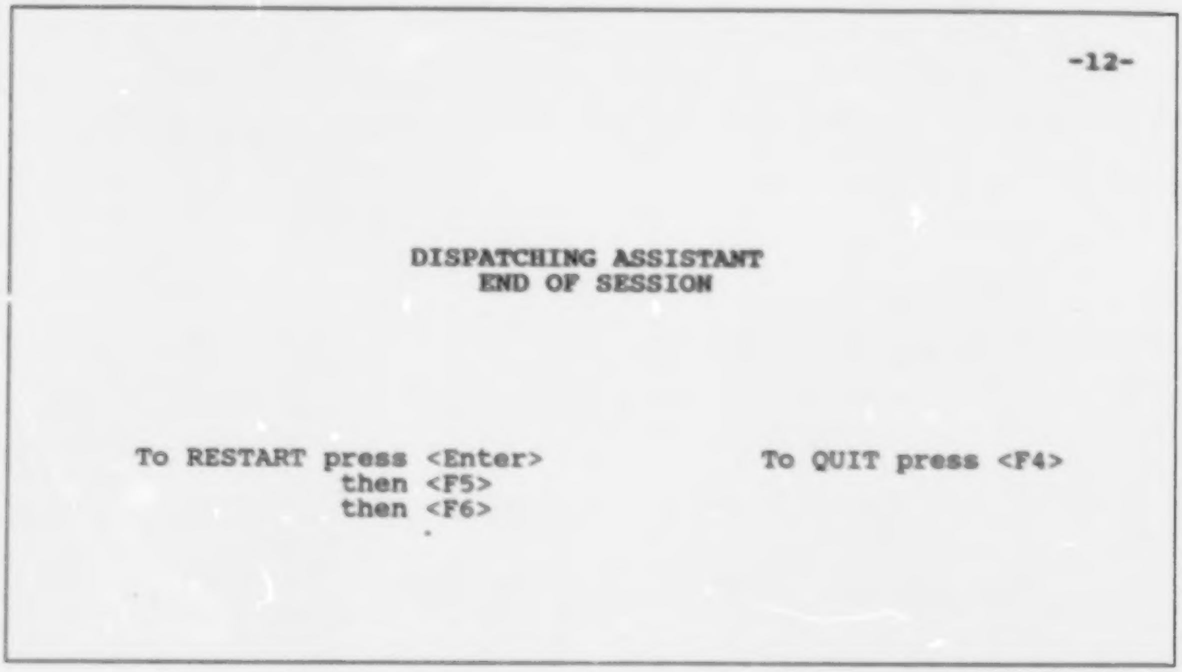
Figure 29 AVAILABLE TRAILERS

```
   customer  zone   gas   use  capacity  kind
************************************************
    STELCOA     2     A    I       3090    f
  TRAILMOBC     1   CO2    I       1100    s
     HOTINC     1   CO2    I       1200    f
 AMSTELBRWC     1   CO2    I       1400    f
   PLANSKYC     1   CO2    I       1209    f
 CAMBLSOUPC     3   CO2    I       2400    f
 STILMEDOWC     3   CO2    I       5290    f
 CLALONDONC     4   CO2    I      13240    f
 CUDDYFOODC     4   CO2    I       9824    s
 FIRESTONEC     4   CO2    I      11630    f
   DELTA70C     4   CO2    I       5500    f
  AJAXHOSPO     1     O    H       2300    f
  COMETECHO     1     O    I       2000    f
 GMBATTERYO     1     O    I       9000    f
  ESCOFAB1O     1     O    I       2300    s
 ESCOFDRY2O     1     O    I       2200    s
 SCARCENTHO     2     O    H       4400    f
 YORKCENTHO     2     O    H       2300    f
  YORKCTYHO     2     O    H       4900    f
  BISHBLDGO     2     O    I       9100    f
  NWESTERNO     2     O    I       4500    f
  ORTHAPEDO     2     O    I        700    f
   CLATOR2O     2     O    I      17600    f
  ETOBHOSPO     2     O    H       4400    f
   PEELHOSPO    2     O    H       9100    f
 WHEATONGLO     2     O    I       9400    f
    EXIDEIO     2     O    I       2300    f
  MISSHOSPO     2     O    H       4400    f
   TONOLLIO     2     O    I       3500    f
  JOSBHOSPO     2     O    H       4600    f
 SCARGRACHO     2     O    H       4500    f
   DOCTORSO     2     O    H       2300    f
  FRANKSTLO     2     O    I       9200    f
 GEOTWHOSPO     2     O    H       1500    f
 OAKVLHOSPO     2     O    H       4600    f
 THERMOBONO     2     O    I       2300    f
   INCORESO     4     O    I      17600    c
 GMSTKITTSO     4     O    I       9800    s
 ELDORADO3N     1     N    I      21600    f
 MOTHERJACN     1     N    I      22200    c
    DUPONTN     1     N    I       7300    s
  PRECRUBBN     2     N    I      15000    f
 SIMCOEINDN     2     N    I       4100    f
  PETCLARKN     3     N    I       2200    s
  HENDHOSPO     3     O    H       4400    f
**************************************************
cust2.nxp  format  89 03 14
```

Figure 30 CUSTOMERS

7.18 END OF SESSION Screen

This screen offers the choice of quitting the Dispatcher's Assistant or starting up a new dispatching session. To return to DOS, press <F4>, and to restart and continue with a new session, press **<Enter>**, followed by <F5> and <F6>. See Figure 31.



```
                                                                -12-




                          DISPATCHING ASSISTANT
                             END OF SESSION




      To RESTART press <Enter>               To QUIT press <F4>
                  then <F5>
                  then <F6>
```

Figure 31 END OF SESSION

7.19 Dragon VoiceScribe

To use the voice interface with the NEXPERT Runtime, a special config.sys file needs to be loaded. A batch file called ac.bat has been created to make this operation easy and efficient. After executing the batch file, reboot the computer.

7.19.1 Training VoiceScribe

The first time a user uses the voice system, he must first train his voice so that it will be recognized. To do this, go to the directory where VoiceScribe is installed and issue the command as follows:

```
    D:>\ cd VIP
```

```
    D:>VIP\Train Dave
```

The list of the words in the vocabulary will appear in alphabetical order and the trainee goes through the list, repeating each word three times. After each word the system tells you if the word has been successfully trained or if it needs to be repeated again. A list of the words in the vocabulary is given in Appendix C.

7.19.2 Using VoiceScribe

Once a voice has been trained on the system, it is then ready to be used. Each time it is used for the first time, it needs to be told who the speaker will be. Type the name at the keyboard.

```
D:>VIP\speaker Dave
```

Voice can then be used to issue the commands, including the command to change directories.

```
D:>VIP\"mot5"
```

The directory will change and the NEXPERT Runtime Protected Mode needs to be installed with the command:

```
D:>MOT5\"setrt"
```

A message will appear on the screen that the protected mode has been installed and the program can be initiated with the command "nx". Execution of the program through voice can proceed as described above.

7.20 **DragonDictate**

7.20.1 Bringing up DragonDictate

For the Dictate to work, the system needs a special config.sys file. Make sure to save your original config.sys file. A batch file, dtc.bat, has been created to replace the present config.sys file with configdt.sys. A complete listing of all the configuration and batch files is included in Appendix B.

```
C:\>dtc
```

A message will appear on the screen to reboot the system for the new config.sys file.

7.20.2 Getting Started

Go to the directory where the work is to be done. For example,

```
D:\>cd ES
```

Load the dictator from D:\DICTATE to the directory. The command line is shown below. Please use your name to load it.

```
D:\ES>dt David
```

A message will appear on the screen that it is loading. When the message window disappears, press the "+" key. If the console window pops up, Figure 32, then you are ready to use the dictator.

7.20.3 Train Vocabulary

Press **"+"** to bring up the console window and select **TRAIN**. After selecting TRAIN, a submenu appears and **TRAIN ALL** should be selected. The system responds with the phrase "Voice Console" on the screen. Read the phrase "Voice Console" five times aloud. If TRAIN ALL is selected the process of repeating the word or phrase five times is repeated until the entire vocabulary is trained. There are 218 instances to be trained. If during a session, a word is read incorrectly, press R to retrain the present word.

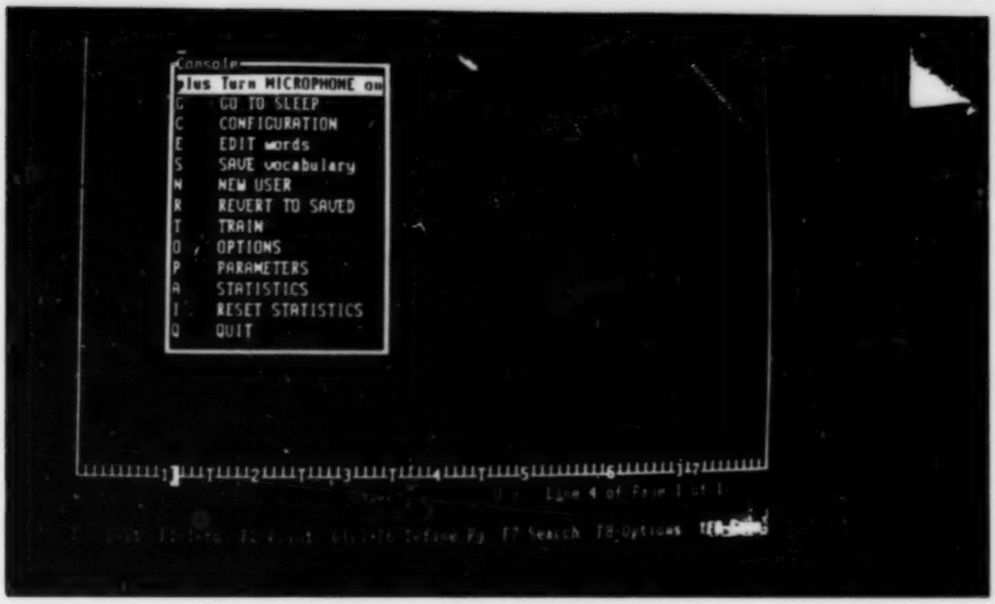

Console
Plus Turn MICROPHONE on
G GO TO SLEEP
C CONFIGURATION
E EDIT words
S SAVE vocabulary
N NEW USER
R REVERT TO SAVED
T TRAIN
O OPTIONS
P PARAMETERS
A STATISTICS
I RESET STATISTICS
Q QUIT
Line 4 of Page 1 of 1
F7-Search F8-Options

Figure 32 DragonDictate Window

7.20.4 Using DragonDictate

The DragonDictate works within an editor, in our case Q&A. To start from the beginning:

```
C:\>dtc
```

Reboot

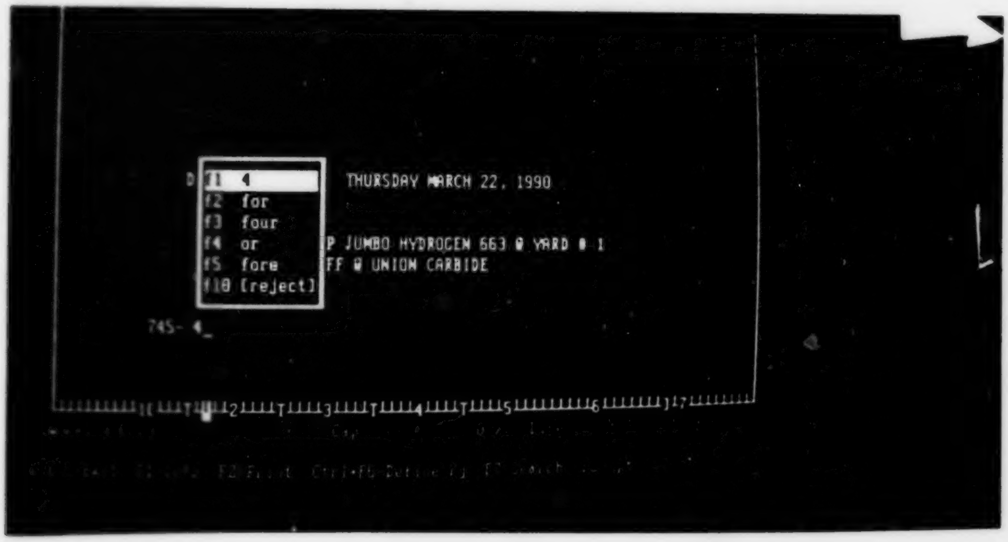
```
C:\>D:

D:>cd DICTATE

D:>QA1
```

Place headphone on head and position the microphone at the corner of your mouth and no further away than 13mm. See Figure 33 for the correct position of the microphone. Press "+" to bring up the console window. A pop-up menu appears and the selection should be **Turn Microphone On**. If the menu shows Turn Microphone Off, then this means the microphone is already on, therefore choose **Q**.

Each time a word is spoken, a choice menu comes up. If the word which has been spoken is in the choice menu then say `Choose' and say the number to correspond with the

Figure 33 Using Dictate Microphone

location of the word, see Figure 34. If the word spoken is the first one on the list then you can go on to the next word. If the word spoken is not in the choice list then say `**Begin spell mode**' to get the window for spelling. Spell the word, using the International Communications Alphabet, which is listed in the DragonDictate User manual, Appendix A, Page 53.



```
D          THURSDAY MARCH 22, 1990

          P JUMBO HYDROGEN 663 @ YARD # 1
          FF @ UNION CARBIDE

745- 4_
```

Figure 34 Selecting Correct Word

The dictator does not respond correctly the first time. Therefore it is recommended that key strokes are used to choose the word or spelling. It is also recommended that the vocabulary be saved at least at the end of a session. This will give better results the next time.

8. Recommendations for Future Developments

This section discusses recommendations to the Transportation Development Centre on the future of the technology developed through the project.

8.1 Linkage with CLA's Prediction System

CLA's dispatch centre in Hamilton, Ontario, currently employs a computerized delivery prediction system. It is assumed that this is also the case in many other similar dispatch centres. To make the over-all scheduling/dispatching process efficient and accurate, it will be necessary to link the dispatch system to the prediction system. The linkage was not intended to be attempted in this project because of the old age of the computer system used to run the prediction system. Severe technical difficulties existed as the old system does not support any standard communication protocols. Nor does it have an easy way to establish a communication channel in any way with the more recent computer selected for the dispatching system (IBM PC AT clone).

In a future project, it would be highly desirable that the two systems be linked or even merged. Then the output of the prediction system can be directly fed into the dispatch system, reducing the chance of introducing errors, and greatly improving the efficiency of data input to the dispatch expert system.

8.2 Importance of Voice-Interactive User Interface

As described in Section 8.4, the extensive knowledge engineering sessions sought by the developers were not possible because of the restricted time made available by the expert designated by CLA for the project. Therefore, a major portion of the resources of the project was spent to investigate the user-interface issues, which are currently the most significant issues in deciding the success/failure of an expert system developed for use in the real world.

The project has made considerable progress in this area, as demonstrated by the number of technical papers and presentations on the project, all invited, at conferences and workshops (Canadian Conference on Electrical and Computer Engineering, September 1989, Montreal;

Transportation Research Board Annual Conference, January 1989, Washington D.C.; L'Heure du CRIM, November 1989, Montreal; DECUS Canada Symposium, January 1990, Toronto; Pacific Rim International Conference on Artificial Intelligence '90, November 1990).

There is a clear indication that the introduction of the voice-interactive front end to a knowledge-based system was the right idea. While there are still improvements to be made, such as an expanded lexicon for faster and more accurate recognition, the eventual benefits of introducing voice-interactive technology to expert systems is clear. It is recommended, therefore, that efforts to develop an effective and practical user interface be continued in order to employ sophisticated computer systems to make transportation efficient and safe.

8.3 Generalization of Methodology

The development so far is highly customized to operations at CLA. In order to make the developed technology applicable to similar operations, it is necessary to generalize both the system architecture and the details of system implementation.

Because of the characteristics of the system development tool (NEXPERT Object) used for the development, the system architecture was kept reasonably general. The modularization of the knowledge base and data bases, and their separation from the rest of the system is also well achieved. The tables and windows used in the user interface can also be applied, in principle, to a number of other similar operations.

It will be desirable, however, to invest additional effort to generalize these components of the system, as well as to make the system architecture somewhat more generic.

8.4 Knowledge Engineering Requirements

Despite the strong support from the management of Canadian Liquid Air, it was extremely difficult to secure the expert's time throughout the project. As a result, the project could not conduct sufficient knowledge engineering. This resulted in a rather small and shallower knowledge base. Unfortunately, it was only after the completion of the prototyping that the development team learned the real reason for the difficulty in gaining access to the expert. In fact, the expert was in a rather precarious position, and was not eager to tip the balance by working on an additional project.

In order to bring the prototype to a system that is useable in the field, it is necessary to conduct further knowledge-engineering sessions with a new expert who would be accessible for knowledge-acquisition sessions.

8.5 Further Development

The final demonstration of the prototype at the dispatch centre in Hamilton, held in February of 1990, revealed that those working in the daily dispatch tasks saw that there was a great potential for such a system to be useful in their actual operation. They offered strong support for further development of the system, thus indicating that the approach taken by the project was viable. It is recommended that further development of the concept and its implementation be considered seriously.

APPENDIX A

Listing of the Rules of the Knowledge Base

```
RULE :  Rule 1
If  there is evidence of database_retrieved
    And there is evidence of driver_change
Then all_shifts_known
    is confirmed.
    And Writing in "lchedule.nxp" @TYPE=NXP;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="3";)

RULE :  Rule 2
If  there is evidence of database_retrieved
    And there is no evidence of driver_change
Then all_shifts_known
    is confirmed.
    And Writing in "lchedule.nxp" @TYPE=NXP;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="3";)

RULE :  Rule 3
If  there is evidence of start
    And Retrieve "lchedule.nxp" @TYPE=NXP;@FILL=ADD;@UNKNOWN=
    TRUE;@CREATE=|driver_class|;
    And Retrieve "ldriver.nxp" @TYPE=NXP;@FILL=ADD;@UNKNOWN=
    TRUE;@CREATE=|driver_class|;
Then database_retrieved
    is confirmed.
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="2";)

RULE :  Rule 4
If  there is evidence of start_scheduling
    And selected_shift is "A","D","N"
    And selected_day is "sun"
    And <|driver_class|>.sun is equal to selected_shift
Then make_schedule
    is confirmed.
    And Create Object <|driver_class|> |selected_drivers|
    And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="'!
    lastname(11)!";@PROPS=tractor,trailer,lastname;@FIELDS
    ="tractor(8)","trailer(11)","lastname(11)";@ATOMS=<|selected
    _drivers|>;   And selected_shift is assigned to |info_class|
    .shift
    And selected_day is assigned to |info_class|.day
    And selected_date is assigned to |info_class|.date
    And Writing in "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)",
    "shift(9)";@ATOMS=|info_class|;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)
```

```
RULE :  Rule 5
If   there is evidence of start_scheduling
     And selected_shift is "A","D","N"
     And selected_day is "mon"
     And <|driver_class|>.mon is equal to selected_shift
Then make_schedule
     is confirmed.
     And Create Object <|driver_class|> |selected_drivers|
     And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
     lastname(11)!";@PROPS=tractor,trailer,lastname;@FIELDS
     ="tractor(8)","trailer(11)","lastname(11)";@ATOMS=<|selected
      drivers|>;   And selected_shift is assigned to
     |info_class|.shift
     And selected_day is assigned to |info_class|.day
     And selected_date is assigned to |info_class|.date
     And Writing in "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''1
     day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)"
     ,"shift(9)";@ATOMS=|info_class|;
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)

RULE :  Rule 6
If   there is evidence of start_scheduling
     And selected_shift is "A","D","N"
     And selected_day is "tue"
     And <|driver_class|>.tue is equal to selected_shift
Then make_schedule
     is confirmed.
     And Create Object <|driver_class|> |selected_drivers|
     And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME
     ="''!lastname(11)!";@PROPS=tractor,trailer,lastname
     ;@FIELDS="tractor(8)","trailer(11)","lastname(11)";@ATOMS
     =<|selected_drivers|>;
     And selected_shift is assigned to |info_class|.shift
     And selected_day is assigned to |info_class|.day
     And selected_date is assigned to |info_class|.date
     And Writing in "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
     !day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)",
     "shift(9)";@ATOMS=|info_class|;
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)
```

```
RULE :  Rule 7
If  there is evidence of start_scheduling
    And selected_shift is "A","D","N"
    And selected_day is "wed"
    And <|driver_class|>.wed is equal to selected_shift
Then make_schedule
    is confirmed.
    And Create Object <|driver_class|> |selected_drivers|
    And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME
    ="''! lastname(11)!";@PROPS=tractor,trailer,lastname
    ;@FIELDS="tractor(8)","trailer(11)","lastname(11)";@ATOMS
    =<|selected_drivers|>;   And selected_shift is assigned to
    |info_class|.shift
    And selected_day is assigned to |info_class|.day
    And selected_date is assigned to |info_class|.date
    And Writing in "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)"
    ,"shift(9)";@ATOMS=|info_class|;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)

RULE :  Rule 8
If  there is evidence of start_scheduling
    And selected_shift is "A","D","N"
    And selected_day is "thu"
    And <|driver_class|>.thu is equal to selected_shift
Then make_schedule
    is confirmed.
    And Create Object <|driver_class|> |selected_drivers|
    And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    lastname(11)!";@PROPS=tractor,trailer,lastname;@FIELDS
    ="tractor(8)","trailer(11)","lastname(11)";@ATOMS=<|selected
    _drivers|>;   And selected_shift is assigned to |info_class|
    .shift
    And selected_day is assigned to |info_class|.day
    And selected_date is assigned to |info_class|.date
    And Writing in "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)"
    ,"shift(9)";@ATOMS=|info_class|;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)
```

```
RULE :  Rule 9
If  there is evidence of start_scheduling
    And selected_shift is "A","D","N"
    And selected_day is "fri"
    And <|driver_class|>.fri is equal to selected_shift
Then make_schedule
    is confirmed.
    And Create Object <|driver_class|> |selected_drivers|
    And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    lastname(11)!";@PROPS=tractor,trailer,lastname;@FIELDS=
    "tractor(8)","trailer(11)","lastname(11)";@ATOMS=<|selected
     drivers|>;
    And selected_shift is assigned to |info_class|.shift
    And selected_day is assigned to |info_class|.day
    And selected_date is assigned to |info_class|.date
    And Writing In "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''1
    day(5)!";@PROPS=day,date,shift;@FIELDS="day(5)","date(13)"
    ,"shift  (9)";@ATOMS=|info_class|;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)

RULE :  Rule 10
If  there is evidence of start_scheduling
    And selected_shift is "A","D","N"
    And selected_day is "sat"
    And <|driver_class|>.sat is equal to selected_shift
Then make_schedule
    is confirmed.
    And Create Object <|driver_class|> |selected_drivers|
    And Writing in "drivtr.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    lastname(11)!";@PROPS=tractor,trailer,lastname;@FIELDS=
    "tractor(8)","trailer(11)","lastname(11)";@ATOMS=<|selected
     drivers|>;
    And selected_shift is assigned to |info_class|.shift
    And selected_day is assigned to |info_class|.day
    And selected_date is assigned to |info_class|.date
    And Writing In "date.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    day(5)!";@PROPS= day,date,shift;@FIELDS
    "day(5)","date(13)","shift(9)";@ATOMS=|info_class|;
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="4";)

RULE :  Rule 11
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "sdriv.txt"
Then printout
    is confirmed.
    And Show "repdriv.txt"
    And Reset print_flag
    And Reset print_report
    And Reset printout
```

```
RULE :  Rule 12
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "trailer.nxp"
Then printout
    is confirmed.
    And Show "trailer.nxp"
    And Reset print_flag
    And Reset print_report
    And Reset printout

RULE :  Rule 13
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "cust2.nxp"
Then printout
    is confirmed.
    And Show "cust2.nxp"
    And Reset print_flag
    And Reset print_report
    And Reset printout

RULE :  Rule 14
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "none"
Then printout
    is confirmed.

RULE :  Rule 15
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "sched.txt"
Then printout
    is confirmed.
    And Show "sched.txt"
    And Reset print_flag
    And Reset print_report
    And Reset printout

RULE :  Rule 16
If  there is evidence of make_schedule
    And there is evidence of print_flag
    And print_report is "sdx.txt"
Then printout
    is confirmed.
    And Show "sdrep.txt"
    And Reset print_flag
    And Reset print_report
    And Reset printout
```

```
RULE :  Rule 17
If  Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="1";)
    And there is evidence of first
Then start
    is confirmed.

RULE :  Rule 18
If  there is no evidence of printout
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="5";)
    And Execute "traildrv.exe"(@TYPE=EXE;@WAIT=TRUE;)
    And Retrieve "traildrv.nxp" @TYPE=NXPDB;FILL=ADD;@NAME="''
    !trailer!";@CREATE=|trailers|;
    And TRUE is assigned to <|trailers|>.avail
    And Retrieve "cust2.nxp" @TYPE=NXPDB;@FILL=ADD;@NAME="''!
    customer!";@CREATE=|customer_class|;
    And there is evidence of no_trail_change
Then B10_no_trail_change
    is confirmed.
    And TRUE is assigned to avail_trail
    And Create Object <|trailers|> |tr_avail|

RULE :  Rule 19
If  there is evidence of printout
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="1";)
    And Execute "traildrv.exe"(@TYPE=EXE;@WAIT=TRUE;)
    And Retrieve "traildrv.nxp" @TYPE=NXPDB;@FILL=ADD;@NAME="
    ''!trailer!";@CREATE=|trailers|;
    And TRUE is assigned to <|trailers|>.avail
    And Retrieve "cust2.nxp" @TYPE=NXPDB;@FILL=ADD;@NAME="''!
    customer!";@CREATE=|customer_class|;
    And there is evidence of no_trail_change
Then B10_no_trail_change
    is confirmed.
    And TRUE is assigned to avail_trail
    And Create Object <|trailers|> |tr_avail|

RULE :  Rule 20
If  there is no evidence of B10_no_trail_change
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="6";)
    And there is evidence of trail_change
    And there is evidence of <|trailers|>.avail
Then B20_change_trail
    is confirmed.
    And Create Object <|trailers|> |tr_avail|
    And TRUE is assigned to avail_trail
```

```
RULE :  Rule 21
If   there is evidence of avail_trail
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="7";)
     And gasX is "O","CO2","N","A"
     And <|customer_class|>.gas is equal to gasX
Then B30_gas_selected
     is confirmed.
     And Create Object <|customer_class|> |cust_gas|
     And TRUE is assigned to <|cust_gas|>.selected
     And TRUE is assigned to b30

RULE :  Rule 22
If   there is evidence of B30_gas_selected
     And <|cust_gas|>.use is "I"
Then B40_calc_ind_required
     is confirmed.
     And 3.0*<|cust_gas|>.capacity/4.0 is assigned to <|cust_gas|>
     .required
     And TRUE is assigned to required_calc

RULE :  Rule 23
If   there is evidence of B30_gas_selected
     And <|cust_gas|>.use is "H"
Then B50_calc_hosp_required
     is confirmed.
     And 2.0*<|cust_gas|>.capacity/3.0 is assigned to <|cust_gas|>
     .required
     And TRUE is assigned to required_calc

RULE :  Rule 24
If   there is evidence of required_calc
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="8";)
     And there is evidence of custsel
     And there is evidence of <|cust_gas|>.selected
Then B60_cust_selected
     is confirmed.
     And Create Object <|cust_gas|> |cust_selected|
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="9";)

RULE :  Rule 25
If   there is evidence of required_calc
     And "1" is assigned to zoneX
     And 1+goto is assigned to goto
     And <|cust_selected|>.zone is equal to zoneX
Then B70_zone1
     is confirmed.
     And Create Object <|cust_selected|> |zone1|
     And <|zone1|>.required+tot_zone1 is assigned to tot_zone1
```

```
RULE :  Rule 26
If  there is evidence of required_calc
    And "2" is assigned to zoneX
    And 1+goto is assigned to goto
    And <|cust_selected|>.zone is equal to zoneX
Then B80_zone2
    is confirmed.
    And Create Object <|cust_selected|> |zone2|
    And <|zone2|>.required+tot_zone2 is assigned to tot_zone2

RULE :  Rule 27
If  there is evidence of required_calc
    And "3" is assigned to zoneX
    And 1+goto is assigned to goto
    And <|cust_selected|>.zone is equal to zoneX
Then B90_zone3
    is confirmed.
    And Create Object <|cust_selected|> |zone3|
    And <|zone3|>.required+tot_zone3 is assigned to tot_zone3

RULE :  Rule 28
If  there is evidence of required_calc
    And "4" is assigned to zoneX
    And 1+goto is assigned to goto
    And <|cust_selected|>.zone is equal to zoneX
Then C00_zone4
    is confirmed.
    And Create Object <|cust_selected|> |zone4|
    And <|zone4|>.required+tot_zone4 is assigned to tot_zone4

RULE :  Rule 29
If  <|tr_avail|>.gas is equal to gasX
Then C10_trailer_gas
    is confirmed.
    And Create Object <|tr_avail|> |tr_gas|
    And zoneX is set to UNKNOWN

RULE :  Rule 30
If  there is evidence of C10_trailer_gas
    And <|cust_selected|>.zone is equal to zoneX
Then C20_write_nxps
    is confirmed.
    And Writing in "custc.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME
    ="''!customer(11)!
    ";@PROPS=required,kind,times;@FIELDS="required(10)","kind(6)"
    "times(7)";@ATOMS=<|cust_selected|>;
    And Writing in "trailc.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME=
    "''!trailer(9)!"
    ";@PROPS=capacity,driver,tractor;@FIELDS= "capacity(10)",
    "driver(12)","tractor(8)";@ATOMS=<|tr_gas|>;
    And 0 is assigned to 'tot_zone'\zoneX\.Value
    And zoneX is set to UNKNOWN
```

```
RULE :  Rule 31
If   there is evidence of C20_write_nxps
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="10";)
     And Execute "scr0222.exe"(@TYPE=EXE;@WAIT=TRUE;)
     And Retrieve "cust3.nxp"
     @TYPE=NXPDB;@NAME="''!customer!";@ATOMS= <|customer_class|>;
Then C30_zone_loop
     is confirmed.
     And Reset C20_write_nxps
     And Reset C30_zone_loop
     And x+1 is assigned to x
     And exe2 is set to UNKNOWN

RULE :  Rule 32
If   there is no evidence of C30_zone_loop
     And <<|tr_sel|>>.driver is UNKNOWN
     And "nn" Is assigned to <<|tr_sel|>>.driver
     And <<<|tr_sel|>>>.tractor is UNKNOWN
     And "xx" is assigned to <<<|tr_sel|>>>.tractor
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="11";)
     And there is evidence of trsel
Then C40
     is confirmed.
     And Writing in "tr3.nxp" @TYPE=NXPDB;@FILL=NEW;@UNKNOWN=TRUE;
     @NAME= "''!trailer(9)!";@PROPS=driver ,tractor;@FIELDS
     ="driver(12)","tractor(8)";@ATOMS=<|tr_sel|>;

RULE :  Rule 33
If   Retrieve "tr3.nxp" @TYPE=NXPDB;@UNKNOWN=TRUE;@SLOTS=tr3.
     trailer, tr3.driver,tr3.tractor;@FIELDS="trailer","driver",
     "tractor";@CURSOR=cursor;
     And <|cust_selected|>.trailer is KNOWN
     And <|cust_selected|>.trailer is equal to tr3.trailer
Then C50
     is confirmed.
     And tr3.driver is assigned to <|cust_selected|>.driver
     And tr3.trailer is assigned to <|cust_selected|>.trailer
     And tr3.tractor is assigned to <|cust_selected|>.tractor
     And Reset C50

RULE :  Rule 34
If   Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="12";)
     And there is evidence of final_change
Then final_check
     is confirmed.
     And Writing in "final.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
     customer(12)!";@PROPS=trailer,driver,tractor;@FIELDS="trail
     er(8)" ,"driver(12)","tractor(8)";@ATOMS=<|cust_selected|>;
     And Execute "printout.exe"(@TYPE=EXE;@WAIT=TRUE;)
     And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="13";)
```

```
RULE :  Rule 35
If  there is no evidence of final_change
Then final_check
    is confirmed.
    And Writing in "final.nxp" @TYPE=NXPDB;@FILL=NEW;@NAME="''!
    customer(12)!";@PROPS=trailer,driver,tractor;@FIELDS="
    trailer(8)","driver(12)","tractor(8)";@ATOMS=<|cust_selected
    d|>;
    And Execute "printout.exe"(@TYPE=EXE;@WAIT=TRUE;)

RULE :  Rule 36
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "gastr"
Then print2
    is confirmed.
    And Show "atrailg.txt"
    And Reset print_flag2
    And Reset print_report2
    And Reset print2
    And final_check is set to TRUE

RULE :  Rule 37
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "final"
Then print2
    is confirmed.
    And Show "finform.nxp"
    And Reset print_flag2
    And Reset print_report2
    And Reset print2
    And final_check is set to TRUE

RULE :  Rule 38
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "none"
Then print2
    is confirmed.

RULE :  Rule 39
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "cust2.nxp"
Then print2
    is confirmed.
    And Show "cust2.nxp"
    And Reset print_flag2
    And Reset print_report2
    And Reset print2
    And final_check is set to TRUE
```

```
RULE :  Rule 40
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "sdriv.txt"
Then print2
    is confirmed.
    And Show "repdriv.txt"
    And Reset print_flag2
    And Reset print_report2
    And Reset print2
    And final_check is set to TRUE

RULE :  Rule 41
If  there is evidence of final_check
    And there is evidence of print_flag2
    And print_report2 is "trailer"
Then print2
    is confirmed.
    And Show "atrail.txt"
    And Reset print_flag2
    And Reset print_report2
    And Reset print2
    And final_check is set to TRUE

RULE :  Rule 42
If  there is no evidence of print2
Then the_end
    is confirmed.
    And Execute "voice.exe"(@TYPE=EXE;@WAIT=TRUE;@STRING="14";)
```

APPENDIX B

Configuration Files Required

The following config.sys files are required for the various functions that are to be carried out.

A .Bat file has been created to make the loading of each of these easy for the user. A listing of the batch files is included here.

<u>CONFIGAC.SYS</u> is the configuration file which needs to be installed to run the Dispatcher's Assistant with voice activated.

```
break = on
buffers = 10
files = 20
device = dmdrvr.bin
device = D:\vip\dragen.sys
device = D:\vip\spkacnt.dvc
```

<u>CONFIGDV.SYS</u> is the configuration file which needs to be installed to use NEXPERT Object, development environment.

```
device=himem.sys
device=c:\qemm.sys
break = on
buffers = 20
device = dmdrvr.bin
files = 20
device=c:\dos\ansi.sys
device=mouse.sys
```

<u>CONFIGRT.SYS</u> is the configuration file which needs to be installed to run NEXPERT Runtime without voice.

```
break = on
buffers = 20
device = dmdrvr.bin
files = 20
```

<u>CONFIGDT.SYS</u> is the configuration file which needs to be installed to use the DragonDictate.

```
break = on
buffers = 10
files = 20
device = qemm.sys
device = dmdrvr.bin
device = d:\dictate\dragon.sys
```

APPENDIX C

Words Trained for Dispatcher's Assistant

The following are the words that have been trained for use with the Dispatcher’s Assistant.

```
_ROOT= assign, available, cancel,
continue, control_q, delete, down_four, down_two,
enter, enter_it,
four, George, go_down, go_left, go_right, go_up, high, low,
March, MOT5, normal, not_available, nx,
one, page_up, page_down, quiet, select, setrt,
show_list, select_four, speak,
one, two, three, four, five, six, seven,
eight, nine, twenty_two, zero,
Hussey, Prevost, Guenther, Burrows, day, night, afternoon;
assign      "’a-a’";
March       "March ";
available   "True";
cancel      "’c-c’";
continue    "’f1’";
control_q   "’c-q’";
delete      "’del’";
down_four   "’down’’down’’down’’down’";
down_two    "’down’’down’";
enter       "’enter’";
enter_it    "’enter’";
four        "4";
George      "George";
Hussey      "Hussey";
Prevost     "Prevost";
Guenther    "Guenther";
Burrows     "Burrows";
go_down     "’down’";
go_left     "’left’";
go_right    "’right’";
go_up       "’up’";
high        "high’enter’";
low         "low’enter’";
MOT5        "mot5’enter’";
normal      "norm’enter’";
not_available "False";
nx          "nx’enter’";
page_down   "’pgdn’";
page_up     "’pgup’";
one         "1";
quiet       "quiet’enter’";
select      " ’down’";
select_four " ’down’ ’down’ ’down’ ’down’";
setrt       "setrt’enter’";
show_list   "’f2’";
speak       "speak’enter’";
three       "3";
five        "5";
six         "6";
seven       "7";
```

```
eight       "8";
nine        "9";
twenty_two   "22";
two         "2";
zero        "0";
day         "D";
night       "N";
afternoon   "A";
```

APPENDIX D

Kepner-Tregoe Decision Analysis

Alternative	EQUIPMENT MANAGEMENT	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	Identifiable skills involved in the process.	Go
Implementable within resource limitations of the project	Feasible if an effective interface to enter updated equipment data can be realized.	Go
Deals with an aspect of dangerous goods transportation	Proper maintenance of the fleet at the company and the vessels installed at customer sites is an essential factor in the safe operation.	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Use of superficial knowledge will likely dominate and the system may be limited in convinciveness.	7	70
Ease of implementation	8	Design of effective data interface will be difficult. Use of simpler explanation methods may be acceptable in this application.	7	56
"Dangerous goods" content	9	The system may provide a tool with which a preventative maintenance schedule can be generated and carried out. Nevertheless, the focus of the act is on the transportation of such goods, not on the maintenance aspect.	7	63
Merits to actual operation	10	Knowledge of vehicles and delivery sites are minor portions of the knowledge involved in planning delivery. Updating site information in daily operation will be a major headache.	5	50

Alternative		EQUIPMENT MANAGEMENT (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	Wt-Sc
Ease of conducting knowledge engineering sessions	7	May have to visit all delivery sites to obtain practical (as opposed to paper-level) descriptions of vessels at the customers. This may prove to be an excessive effort.	5	35
Self-containedness of the system	7	Site information on HP-250 computer. Equipment readiness information also must be updated regularly.	5	35
Expected completeness of implementation within resource constraints	9	It is likely that not all equipment data can be covered adequately. Site information particularly hard to cover and maintain.	5	45
Ease of selling the concept	6	Equipment management not the heart of the scheduling operation.	7	42
Applicability to other similar operations	8	Equipment, maintenance requirements differ widely from operation (e.g., milk delivery) to operation (e.g., fire station). Yet the management of equipment is a popular domain for which expert systems have been built. The system may be introduced to other operations of the company.	7	56
Expendability of the system	7	Data interface can be improved so that more can be automated for the system.	7	49
		Total Weighted Score		501

Alternative	SCHEDULE GENERATION ADVISOR	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	Since the process of scheduling at the company is very time consuming (up to 3 hours) and complex, (currently, only the expert can handle it), a successful implementation of the system will be very convincing.	Go
Implementable within resource limitations of the project	By limiting the depth to which details of the interacting knowledge sources is treated in the prototype, an effective prototype may be developed.	Go
Deals with an aspect of dangerous goods transportation	By emphasizing the need for drivers to operate within legal requirements for loading, transporting, and unloading, points can be made in regards to the Act. Also, "good scheduling makes a safer transportation".	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Drastic reduction in scheduling time and effort spent by the expert may be achieved.	10	100
Ease of implementation	8	Must extract most minute pieces of knowledge from the expert. The selection of approapriate representation may be a problem for some type of knowledge. The scale of the development will be quite substantial and the effort to trim it to the the scale acceptable for the allocated resource may prove difficult. The coordinated synthesis of subsystems of the prototype may also be straneous because of the anticipated size of the knowledge base.	6	48

Alternative		SCHEDULE GENERATION ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
"Dangerous goods" content	9	Represented only indirectly. Matters and issues related to the Act can be emphasized in various locales of the system. However, they will all be secondary and thertially in importance.	6	54
Merits to actual operation	10	Will be a useful system if well balanced in functions and well packaged. Possibility of creating a weak prototype is a reality. The framework of the development will become a good template for similar scheduling problems.	10	100
Ease of conducting knowledge engineering sessions	7	Expected to be extensive and highly complicated. Volume of work will be large. The expert has difficulty articulating skills used in certain decision making actions he undertake (as usually is the case with experts). Furthermore, he has difficulty recognizing the significance of some of the steps he routinely uses in his job.	6	42
Self-contained-ness of the system	7	Must get information from HP-250 to be truly effective. This is not a trivial process, and may even prove totally impractical. An option then is to create a local set of databases in addition to the knowledge base. The update problem will then become a major headache to deal with. It is going to be messy one way or another.	6	42
Expected completeness of implementation within resource constraints	9	Useful system can be constructed, if ways to balance depth of knowledge found and the data interface problem solved. It is easy to expand segments of knowledge, exceeding resource limitations.	5	45

Alternative		SCHEDULE GENERATION ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Ease of selling the concept	6	Since the system will take over a major task, it is easy to sell.	10	60
Applicability to other similar operations	8	Can be applied to other distribution centres of the company. Not likely applicable to other dispatch operations because of intricate meshing of domain-specific and generic knowledge.	7	56
Expandability of the system	7	Voice i/o will likely be highly successful.	9	63
		Total Weighted Score		610

Alternative	SCHEDULE ALTERATION ADVISOR	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	Similar in complexity to SCHEDULE GENERATION ADVISOR, but not as dramatic in effect.	Go
Implementable within resource limitations of the project	Yes, by limiting the scope of the project.	Go
Deals with an aspect of dangerous goods transportation	Limited, but yes.	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Less time saving than the SCHEDULE GENERATION ADVISOR. Still the effect of the prototype will be visible.	8	80
Ease of implementation	8	Effective user interface most important, and it is not easy to come up with one for this application. Application of voice input/output will be effective, but again, not easy to develop a robust and effective subsystem.	6	48
"Dangerous goods" content	9	Only in terms of "a good dispatch makes a safer operation".	6	54
Merits to actual operation	10	Will be useful, but frequency of its usage may not be high.	7	70
Ease of conducting knowledge engineering sessions	7	Expected to be as extensive and complicated as in SCHEDULE GENERATION ADVISOR.	6	42

Alternative		SCHEDULE ALTERATION ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Self-containedness of the system	7	Must get information from HP-250 to be truly effective. This will be very difficult to achieve.	6	42
Expected completeness of implementation within resource constraints	9	Smaller than SCHEDULE GENERATION ADVISOR, but still involves similar complexity in design, knowledge engineering, and implementation.	6	54
Ease of selling the concept	6	Deals with only a minor aspect of the operation. However, the prototype's ability to deal with complex rescheduling problems may attract interest.	8	48
Applicability to other similar operations	8	Techniques and skills developed will be useful but the actual system not usable in other applications.	6	48
Expandability of the system	7	Could be made into combined schedule generator/modifier by incorporating functions needed in the SCHEDULE GENERATION ADVISOR. Addition of voice interface can be effective.	8	56
		Total Weighted Score		542

Alternative	VEHICLE OPERATION ADVISOR	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	Yes. The complex sequence of events required to ready vehicles may be modeled and effective advice given based on the model.	Go
Implementable within resource limitations of the project	Highly complex process but feasible because of relatively limited number of trailers and tankers.	Go
Deals with an aspect of dangerous goods transportation	Proper preparation and maintenance of vehicles is extremely important for safe transportation of dangerous goods.	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Somewhat limited in effect because vehicle preparation is a relatively small sub-task of the over-all scheduling task. A high degree of success may be achieved by manual systems. May be difficult to compete with such systems.	7	70
Ease of implementation	8	Again, ability to access up to date vehicle information is most crucial in making the system a success. This goal is not easily achievable and the mechanism to support such an update will be quite involved.	9	72
"Dangerous goods" content	9	Safe condition of the vehicle, tank, and the equipment for loading and unloading are all essential for a safe operation of the centre, and though not directly, relate to aspects of the Act.	8	72

Alternative		VEHICLE OPERATION ADVISOR (Continued)		
WANT Conds.	**Wt.**	**Information/Observation/Analysis**	**Sc.**	**WtSc.**
Merits to actual operation	10	Will be quite useful, if successful. Will be used frequently. Nevertheless, the percentage of equipment maintenance in the entire scheduling operation is still small and the over-all effect of the prototype may be questioned.	8	80
Ease of conducting knowledge engineering sessions	7	More systematic. Model construction will be challenging technically. Will differ substantially from conventional knowledge engineering activities.	8	56
Self-containedness of the system	7	Still need external information regarding the entire fleet of vehicles. Nevertheless, the extent of external interface is rather limited compared to others (e.g., vessel maintenance) and can be implemented and maintained within a reasonable amount of effort.	8	56
Expected completeness of implementation within resource constraints	9	Useful system can be constructed, if ways to achieve efficient vehicle info update. The limitation of domain should work positively.	7	63
Ease of selling the concept	6	Sellable as a practical "dangerous-good" application, though not generic in nature.	10	60
Applicability to other similar operations	8	Good entry to dangerous goods advisor application. Because site-specific information will be used heavily, direct transportation of the system to some other operation will not be easy.	8	64

Alternative		VEHICLE OPERATION ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Expandability of the system	7	Addition of voice input/output may be a good area to pursue. The system can be incrementally refined towards a highly sophisticated intelligent advisory system.	10	70
		Total Weighted Score		663

Alternative	LEGAL COMPLIANCE ADVISOR	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	A rich and active area of R&D in expert systems.	Go
Implementable within resource limitations of the project	Dangerous Goods Act is huge. However, it can be easily dissected into components and each of them separately implemented.	Go
Deals with an aspect of dangerous goods transportation	Yes.	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Fairly sophisticated questions may be answered by the advisor, effectively demonstrating the power of the approach.	8	80
Ease of implementation	8	Treatment of legal precedents is an area which is still very much in its infancy. Without the ability to reason about past similar cases the performance of the system will be limited. The statutory aspects of the Act may be implemented relatively easily. The truly effective implementation of the Act will involve methods to access the huge database, which is an important part of the Act.	10	80
"Dangerous goods" content	9	Very high. The system will be mostly about the Act.	10	90

Alternative		LEGAL COMPLIANCE ADVISOR		(Continued)
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Merits to actual operation	10	Useful if sufficiently sophisticated. It is anticipated a marginal level of proficiency can be achieved within the confines of the resources allocated for the project. The legal aspects are rather fixed within the entire scheduling operation. Once issues are identified and solutions developed (signs, procedures, hours, communications, etc.), they do not impact the day to day dispatching of the operation, except, possibly in emergencies.	6	60
Ease of conducting knowledge engineering sessions	7	Can be carried out more or less independent of the expert, through the documentation on the Act. Representation of some legal concepts could be difficult.	9	63
Self-containedness of the system	7	Can be built totally independent of the HP-250 computer.	10	70
Expected completeness of implementation within resource constraints	9	An example of a useful legal expert system can be constructed and demonstrated. The prototype will give a basis from which the cost of implementation of other sections of the Act can be calculated.	9	81
Ease of selling the concept	6	Interest on implementation of Dangerous Goods Act currently very high and expected to remain high.	10	60

Alternative		LEGAL COMPLIANCE ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Applicability to other similar operations	8	Can be applied to other distribution centres of the company and other companies involved in transporting liquid gases. The framework of the system will be applicable to a wide range of legal expert system applications.	10	80
Expandability of the system	7	Readily expandable to other sections of the Act. Can be expanded to assist with the paperwork involved and generation of required reports.	10	70
		Total Weighted Score		734

Alternative	DRIVER SCHEDULING ADVISOR	
MUST Conditions	Information/Observation/Analysis	Go/No
Demonstrates effectiveness of expert system	Yes. In particular, the fairness aspect of the system may be attractive to both the dispatcher and the drivers.	Go
Implementable within resource limitations of the project	Will fit in budgeted resource allocation with relative ease.	Go
Deals with an aspect of dangerous goods transportation	Experience and qualifications of each driver can be emphasized during scheduling. Also, the "good scheduling makes safer transportation" aspect.	Go

WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Will visibily unload dispatcher's workload by taking over more sensitive but un-rewarding aspects of the job. Will be able to demonstrate the fairness aspects. This will be important since the expert is interested in assigning a less experienced individual to the dispatching job.	9	90
Ease of implementation	8	This is an area in which the expertise of the current scheduler is most prominent. Because of this it may not be easy to create a system which is comparable in its capability.	8	64
"Dangerous goods" content	9	Experience and qualifications of each driver may be highlighted in the design and implementation. Still a relatively weak argument.	6	54
Merits to actual operation	10	A high potential for developing a truly useful operational aid exists. It will be highly useful for a less experienced person. Nevertheless, the system will cover only a part of the over-all scheduling task.	9	90

Alternative		DRIVER SCHEDULING ADVISOR	(Continued)	
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Ease of conducting knowledge engineering sessions	7	While the activity will be contained in a small subdomain, very extensive exchanges with the expert will be required. Representation of some of his knowledge may not be straightforward.	7	49
Self-containedness of the system	7	It is most likely that access to the HP-250 computer is not necessary to process information concerning the drivers. Even if some of the information has to be reproduced on the AT, the size of the fleet is small enough to justify the duplication.	10	70
Expected completeness of implementation within resource constraints	9	A high degree of completeness and sophistication will likely be achieved within the resources allocated for the project. However, this does not imply the finished prototype can be readily used in the operational environment without further engineering refinements.	10	90
Ease of selling the concept	6	Since the system will actually assist in the selection of drivers, the selling will be easy.	9	54
Applicability to other similar operations	8	Can be applied to other distribution centres of the company. Personnel management systems are in great demand throughout the industry. Further refinements and eventual packaging are possible because of the general nature of the problem.	9	72

Alternative		DRIVER SCHEDULING ADVISOR (Continued)		
WANT Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Expandability of the system	7	Expansions will be made inward. Knowledge structures and the implementation of the knowledge base can be improved incrementally. It will be a while before the system grows enough to demonstrate human-like qualities (sensitivity to needs of individual drivers, spotting of potential problems, advice generation). Refinements and expansions can be made to the explanation facility. Additional MIS hookups (logging, report generation, etc.) can be developed.	9	63
Total Weighted Score		696		

Alternative	MAINTENANCE SCHEDULING ADVISOR	
MUST Conditions	**Information/Observation/Analysis**	**Go/No**
Demonstrates effectiveness of expert system	Yes. The prototype will be able to generate useful maintenance advice regularly.	Go
Implementable within resource limitations of the project	Shall be able to create a useful system despite the large number of customer sites involved.	Go
Deals with an aspect of dangerous goods transportation	Proper maintenance of vessels is always an important aspect in the treatment of liquid gas, and therefore contributes to safer transportation of such gases.	Go

Want Conds.	Wt.	Information/Observation/Analysis	Sc.	WtSc.
Effectiveness of demonstration	10	Somewhat limited in dramatic effect, though still generates useful results.	7	70
Ease of implementation	8	Mechanism to generate advisory information may need further study. Effective representation of vessels may also take time. However, this effort will be well rewarded due to its pioneering nature.	9	72
"Dangerous goods" content	9	High. Well-kept storage will be a key component of a safe operation.	8	72
Merits to actual operation	10	High potential for developing a truly useful operational aid. However, its impact on the scheduling itself will be limited.	6	60
Ease of conducting knowledge engineering sessions	7	Almost no need to interview the scheduling expert. Interviews with maintenance experts both at CLA and the client sites will become necessary. This may affect the plan of development.	7	49